APPLICATION OF A FINITE ELEMENT METHOD TO THE BAROTROPIC PRIMITIVE EQUATIONS

Donald Ernest Hinsman

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

APPLICATION OF A
FINITE ELEMENT METHOD TO THE
BAROTROPIC PRIMITIVE EQUATIONS

by

Donald Ernest Hinsman

September 1975

Thesis Advisors: G.J. Haltiner
R.T. Williams

Approved for public release; distribution unlimited.

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Application of a Finite Element Method to the Barotropic Primitive Equations		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis; September 1975 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Donald Ernest Hinsman		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		12. REPORT DATE September 1975 13. NUMBER OF PAGES 116
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Barotropic Primitive Equations
Finite Element Method
Icosahedral Grid

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A finite element application to the barotropic primitive equations is presented including theoretical development and the model used. Analytic initial data is generated in order to verify as well as possible the accuracy of the model. A comparison of the model with similar finite difference schemes shows that this finite element method exhibits better phase speed propagation than comparable second and fourth order

(20. ABSTRACT Continued)

finite differencing and is competitive in the size of the allowable time step.

Application of a Finite Element Method to the Barotropic Primitive Equations

by

Donald Ernest Hinsman
Lieutenant, United States Navy
B.S., United States Naval Academy, 1968

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN METEOROLOGY

from the

NAVAL POSTGRADUATE SCHOOL
September 1975

ABSTRACT

A finite element application to the barotropic primitive equations is presented including theoretical development and the model used. Analytic initial data is generated in order to verify as well as possible the accuracy of the model. A comparison of the model with similar finite difference schemes shows that this finite element method exhibits better phase speed propagation than comparable second and fourth order finite differencing and is competitive in the size of the allowable time step.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF CHARTS

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

A_1	Area of a smaller triangle in area coordinates
A_2	Area of a smaller triangle in area coordinates
A_3	Area of a smaller triangle in area coordinates
A^*	Arbitrary constant in the stream function
$\overline{A}$	Matrix used in equation formulation
A	Area of triangle
A_m	Arbitrary constant for the Fourier series cosine terms
$A(\theta)$	Constant in analytic initial geopotential
A_o	Arbitrary constant for the Fourier series
a	Earth's radius
a_1	Natural coordinate definition
a_2	Natural coordinate definition
a_3	Natural coordinate definition
B	Constant in the stream function
B_m	Arbitrary constant for the Fourier series sine terms
$\overline{B}$	Matrix used in equation formulation
$B(\theta)$	Constant in analytic initial geopotential
b	Constant in basis function
b_1	Natural coordinate definition
b_2	Natural coordinate definition
b_3	Natural coordinate definition
C_o	Arbitrary constants for the Fourier series combined terms
$C(\theta)$	Constant in analytic initial geopotential
C_m	Arbitrary constant for the Fourier series

c	Phase speed of the fastest gravity wave
$\overline{D}$	Matrix used in equation formulation
d	Constant in basis function
$\overline{E}$	Vector used in equation formulation
e	Constant in basis function
$\overline{F}$	Matrix used in equation formulation
f	Coriolis parameter
$\overline{G}$	Vector used in equation formulation
L_1	Area coordinate definition
L_2	Area coordinate definition
L_3	Area coordinate definition
M	Wave number plus one
m	Zonal wave number
NACA	National Advisory Committee on Aeronautics
N	Number of points in icosahedral grid, Number of unknowns in matrix equation
n	Number of segments each major spherical triangle's side is divided into
T	Number of triangles in icosahedral grid
t	Time
Δt	Time increment
u	Zonal wind
u^*	Second order approximation in time
v	Meridional wind
v^*	Second order approximation in time
Δx	Distance increment in the x-direction
V	Basis function
V_i	Pyramid function

α	Coefficient for the representation of u
α^*	Second order approximation in time
β	Coefficient for the representation of v
β^*	Second order approximation in time
δ_m	Phase angle for wave number m
∇^2	Laplacian operator
$< \ >$	Inner product definition
$\underline{\leq} \ \underline{\geq}$	Inner product definition
η	Coefficient for the representation of sin θ
λ	Longitude
ν	Angular wave velocity
ξ	Coefficient for the representation of cos θ
θ	Latitude
Φ	Geopotential
ψ	Stream function
Ω	Angular velocity of the earth
∇	Del operator
$\frac{\partial}{\partial t}$	Partial time derivative
$\frac{\partial}{\partial \lambda}$	Space derivative in the λ direction
$\frac{\partial}{\partial \theta}$	Space derivative in the θ direction
π	Constant equal to 3.1415926
γ	Coefficient for the representation of ϕ
γ^*	Second order approximation in time

ACKNOWLEDGEMENTS

The author would like to thank the many people from the teaching staff at the Naval Postgraduate School who contributed to this thesis. In particular, to Dr. G. J. Haltiner for his patience and guidance in preparing this paper, to Dr. R. T. Williams for his encouragement and advice, to Dr. C. Comstock for his ever open door and to Dr. D. Salinas for his help with all the technical aspects of the finite element method. To Dr. D. Archer, the author can only say that without his knowledge and help this thesis would never have been completed. Finally, I give my love to my wife who has endured and encouraged me throughout my graduate studies at the Naval Postgraduate School.

I. INTRODUCTION

With the advent of the computer, there have been tremendous advances in the field of numerical weather prediction. Although there are several methods available for solving the prediction equations, the main thrust of the development of the numerical models has been with the finite difference method. Parallel developments in the engineering fields have used other methods as well as finite difference schemes. One recent technique being used is the finite element method. The purpose of this paper was to develop a finite element application to a barotropic primitive equation model. The objective was to learn the characteristics of a finite element model and compare it with a similar finite difference model. Analytic initial conditions were used to insure the most accurate analysis possible and to simplify the comparisons.

II. FINITE ELEMENTS

Although the finite element method has only recently emerged as an efficient means of solving differential equations, its beginnings can be traced loosely back to early times. The basic concept of the finite element method is that a solution can be accurately determined by a sum of simple, easily computable functions. This procedure is not new. Martin (1973) states that a Chinese engineer in A. D. 480 was able to determine upper and lower bounds on the value of π to seven digits. This was accomplished by accurately determining the area of a circle with slender inscribed and circumscribed polygons. From calculus of variations, the classical Rayleigh-Ritz method shows how to approximate a solution to certain problems with a linear combination of any set of linearly independent functions. The Rayleigh-Ritz method determines the weights that are associated with each function. For a great many years both engineers and mathematicians were hung up on the idea that each of these functions must be essentially non-zero over the entire domain of the problem. For a large complex solution, the determination of the weights could be prohibitive. Then in the early 1950's engineers applied the Rayleigh-Ritz method subdividing the entire domain into many smaller pieces or elements. Hence the anachronism, finite elements. In this respect, the spectral method and the finite element method

are quite similar. The spectral method is a combination of sines and cosines defined over the entire domain while the finite element method is a combination of low order polynomials, each polynomial or basis function, being non-zero only over a small "finite element." The question is then how to choose the coefficients for the polynomial to best approximate the answer to the equation. The classical Rayleigh-Ritz procedure for linear self-adjoint problems formulates the problem as the solution to a minimization of a positive definite functional. The coefficients then are chosen to minimize the error in using a finite number of terms. Galerkin proved that the same coefficients result if the problem is formulated as a self-adjoint linear differential equation and then the coefficients are chosen in the following manner. Insert the linear combination of N basis functions into the equation, multiply this resulting equation by N linearly independent test functions (usually the basis functions themselves) and integrate to get N equations for the N coefficients. (Observe that if the test and basis functions are the same, multiplying the basis and test functions involves the integral of the squares of the functions, and thus this procedure is related to the least square error – a fact which Galerkin proved.)

One may suspect that this same procedure would also give a reasonable error for nonlinear, non-self adjoint problems.

In summary, the Rayleigh-Ritz-Galerkin procedure for nonlinear, non-self adjoint equations involves subdividing

the domain into a set of elements, approximating the dependent variables by a linear combination of low order polynomials (having value zero except over the particular element), inserting these approximations into the equation, multiplying the equation by a test function (also a low order polynomial having a one-to-one correspondence with the basis functions) whose purpose is to minimize the residual between the approximate solution and the actual solution, integrating over the entire domain, and finally solving the system of equations assembled during the integration for the solution. Since the basis functions are zero over almost all of the area over which integration takes place, this procedure produces a system of equations for which the matrix is very sparse. The Rayleigh-Ritz-Galerkin method has become the most popular finite element method and is used in this paper. Martin (1973), Zienkiewicz (1971), Strang (1973), Schultz (1973), Norrie (1973), and Aziz (1972) give in-depth theoretical descriptions of the finite element method.

The first step in the Rayleigh-Ritz-Galerkin method requires the division of the domain into a set of elements. The element shape in this paper was chosen to be triangular. Section III contains a description of the subdivisions utilized.

The next step requires representing the dependent variables as a linear combination of low order polynomials. In this paper the polynomials used for both the test and basis functions were <u>linear</u> in λ and θ. A linear basis function

over a particular element can be visualized as a plane with value one at one of the vertices of the triangle, value zero at the other two vertices of the triangle and identically zero over the rest of the domain. Since any one element has three points, there are three planes used in approximating a function over each element. This can be seen in Figure 1. Any function, say u, can be represented over the particular element as a linear combination of the three planes shown in Figure 1 by the equation

$$u_{element} = \sum_{j=1}^{3} \alpha_j(t)\ V_j \qquad \text{(II-1)}$$

where α_j represents the scalar value of u at point j of the element and V_j are the basis functions. At the boundary of two adjacent elements, the dependent variables are continuous. It may be noted, however, that since only linear functions are used for the representation, derivatives are not continuous along the boundaries.

After the approximation for the variables have been substituted into the equations, the next step involves multiplying by a test function. Since in this paper, the test and basis functions were the same, Figure 1 also shows the three planes making up the test functions over any one element.

The next step in the Rayleigh-Ritz-Galerkin procedures is to integrate over the domain. Since each basis and test function is zero over the domain except over the particular

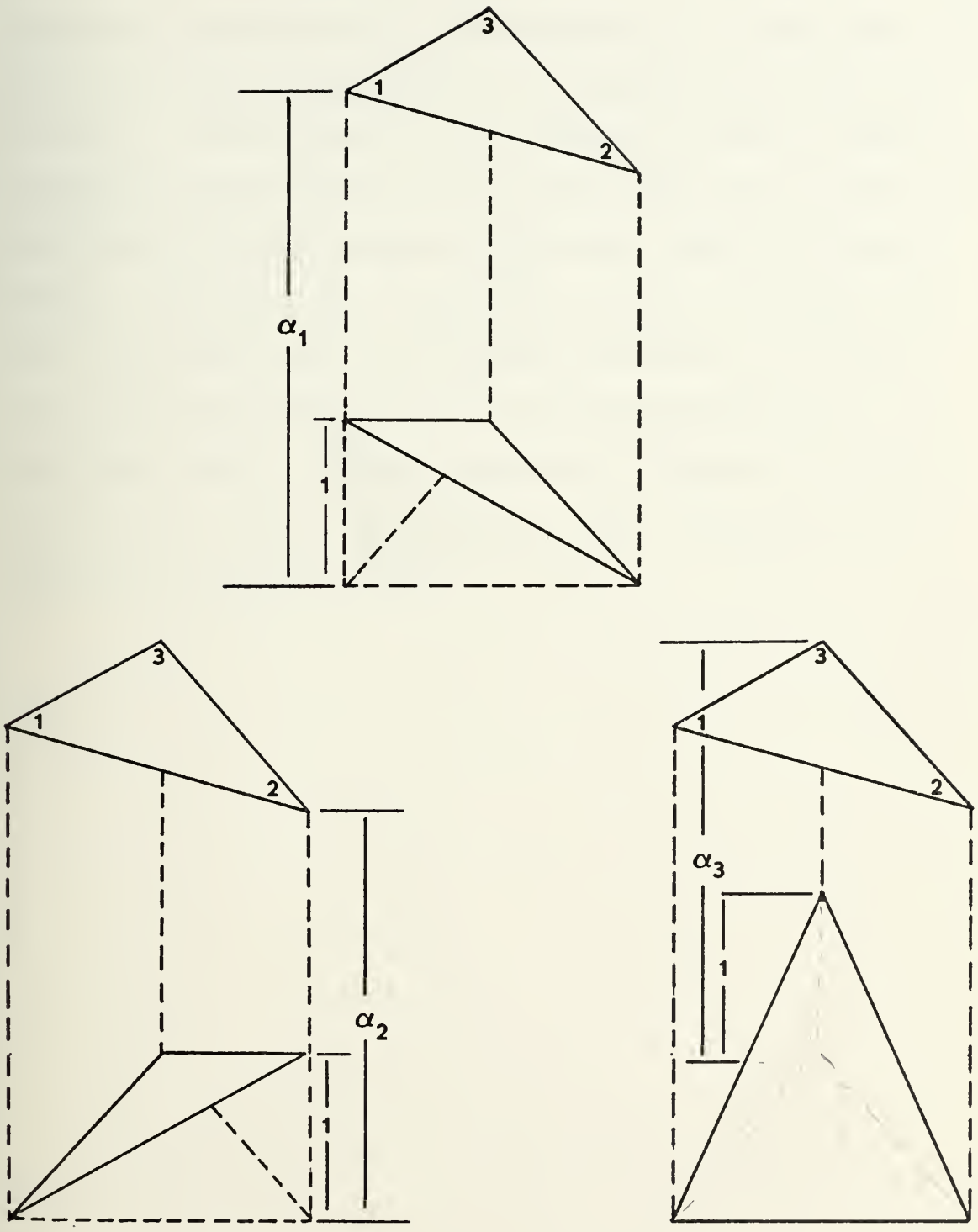

FIGURE 1. Three planes representing a function over an element.

element, the global integration can be performed by integrating locally over each element. In this manner the equations are written and then solved for at each node. The basis and test functions may be visualized from the viewpoint of a node instead of an element as shown in Figure 2 where the six basis functions having value one at point i,j for each of the six elements are shown. The six planes (basis functions) shown make up a "pyramid function" for the grid point i,j. This "pyramid" function is the test function that multiplies the variable representation as described above. If this procedure is repeated for the N grid points a system of equations with N equations and N unknowns will be generated.

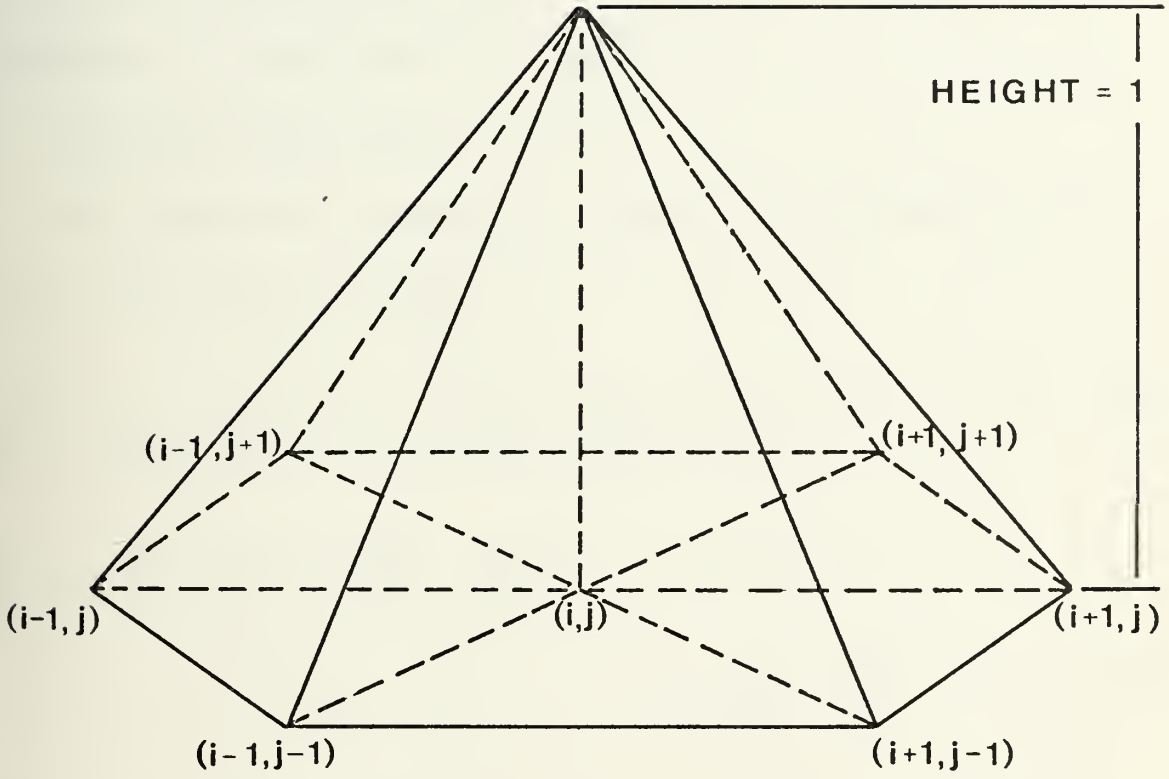

FIGURE 2. Pyramid function at point i,j

III. BAROTROPIC PRIMITIVE EQUATION MODEL

The time integration was performed using a Galerkin finite element application to the so called shallow water, "primitive" equations. A time extrapolated Crank-Nicholson method was used during formulation. The Crank-Nicholson method is explained in some detail in Appendix A.

A. PRIMITIVE EQUATIONS

The primitive equations, in spherical coordinates, for this model are as follows:

$$\frac{\partial \phi}{\partial t} + \frac{1}{a \cos \theta} \frac{\partial}{\partial \lambda} (\phi u) + \frac{1}{a \cos \theta} \frac{\partial}{\partial \theta} (\phi v \cos \theta) = 0 \qquad \text{(III-1)}$$

$$\frac{\partial u}{\partial t} + \frac{u}{a \cos \theta} \frac{\partial u}{\partial \lambda} + \frac{v}{a} \frac{\partial u}{\partial \theta} - \frac{uv}{a} \tan \theta - 2\Omega \sin \theta \, v + \frac{1}{a \cos \theta} \frac{\partial \phi}{\partial \lambda} = 0 \qquad \text{(III-2)}$$

$$\frac{\partial v}{\partial t} + \frac{u}{a \cos \theta} \frac{\partial v}{\partial \lambda} + \frac{v}{a} \frac{\partial v}{\partial \theta} + \frac{u^2}{a} \tan \theta + 2\Omega \sin \theta \, u + \frac{1}{a} \frac{\partial \phi}{\partial \theta} = 0 \qquad \text{(III-3)}$$

After equations (III-1), (III-2), and (III-3) were expressed in finite element form, they were solved in the following order: the continuity equation (III-1), the u-equation (III-2), and the v-equation (III-3). Each equation

was solved by an iteration procedure until a converged solution was achieved before proceeding to the next equation.

B. GRIDS

Two different grids were used for experiments during development of the computer model, namely, the simple λ,θ grid and the icosahedral grid. The λ,θ grid conserves angular distance in radians while the icosahedral grid nearly conserves linear distance in meters.

1. λ,θ Grid

Since the primitive equations are normally expressed in the spherical coordinate form, it seemed natural to use a grid simply subdivided by longitude and latitude. Longitude ranged from zero to 2π and latitude, from $-\frac{\pi}{2}$ to $\frac{\pi}{2}$. The grid was then evenly subdivided into the desired intervals. A ten-degree interval generated a "square" with a side of length ten degrees. The construction of one diagonal across each square further subdivided each square into two triangles with the same square radian area. (See Figure 3.) The intersection of adjacent triangles' apices defined a node. Each node was then assigned a global correspondence number. The numbering scheme ran along constant latitude lines in order to keep the bandwidth of the coefficient matrix as narrow as possible. Cyclic continuity was maintained by numbering the nodes at $\lambda = 2\pi$, with the same number as the $\lambda = 0$ node given the same latitude circle. Every internal

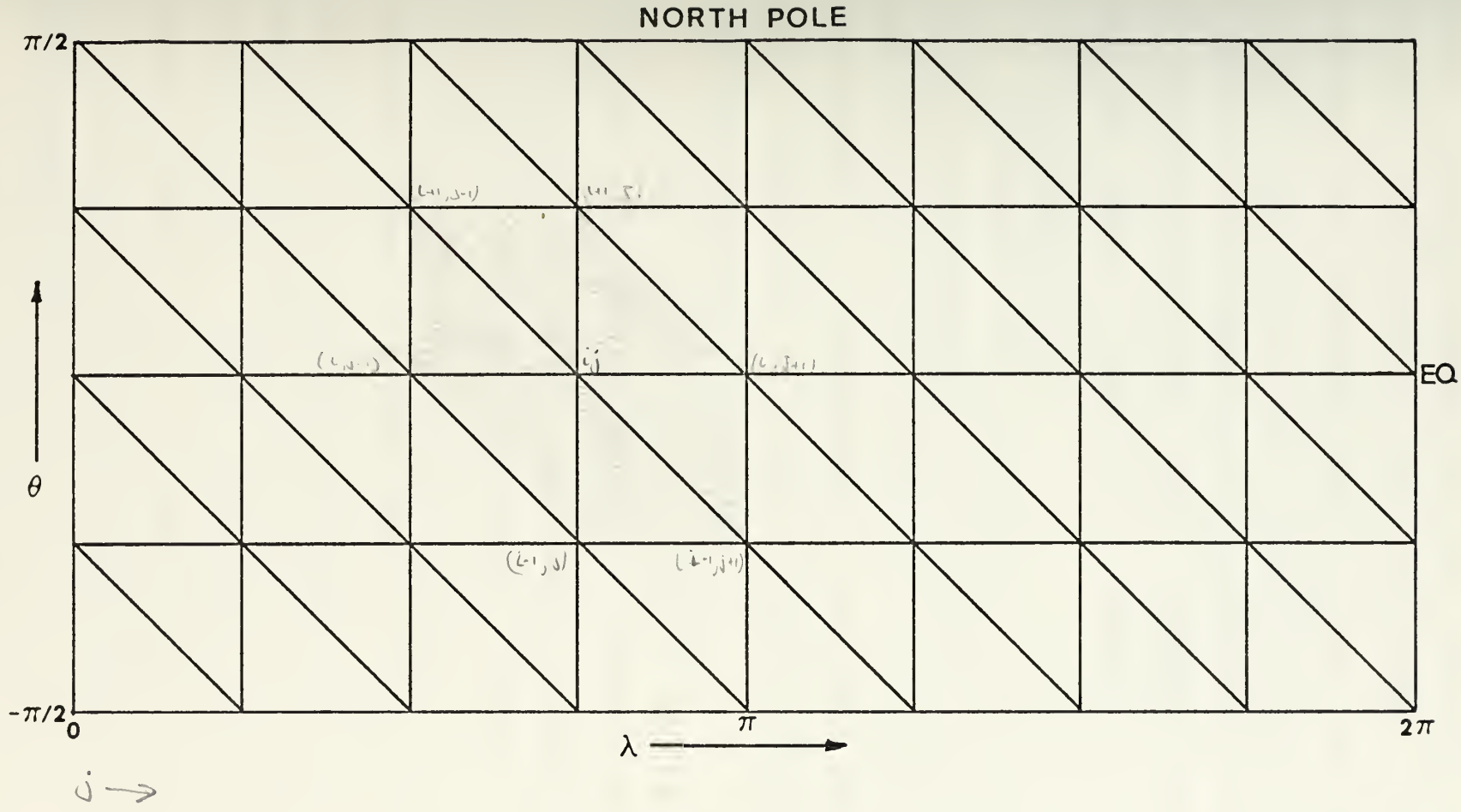

FIGURE 3. λ,θ grid

node was supported by six other nodes; the north and south pole nodes were supported by four other nodes. A drawback to this grid was the north and south poles being represented by a line rather than a node. The primitive equations in the spherical coordinate form are singular at the poles. Thus this grid placed many nodes at one point (the north or south pole) where the equations are singular.

2. Icosahedral Grid

Williamson (1968) and Sadourny, et al. (1967) pointed out the advantages of a grid which is nearly homogeneous with respect to area, and both described methods of generating a grid which used an icosahedral spherical figure. Cullen (1974) also used a regular icosahedron while integrating the primitive equations.

A regular icosahedron on a sphere consists of twenty major spherical triangles with twelve vertices (see Figure 4). An icosahedral grid is then superimposed by subdividing the major triangles into smaller triangles of nearly uniform area, which is the most important feature of this grid. If a model can be run on a global grid with nearly equal area triangles then the problem of decreasing time steps associated with decreasing Δx as the poles are approached can be eliminated. Cullen (1974) subdivided each major spherical triangle along latitude/longitude circles but pointed out that a great circle subdivision would best conserve equal areas.

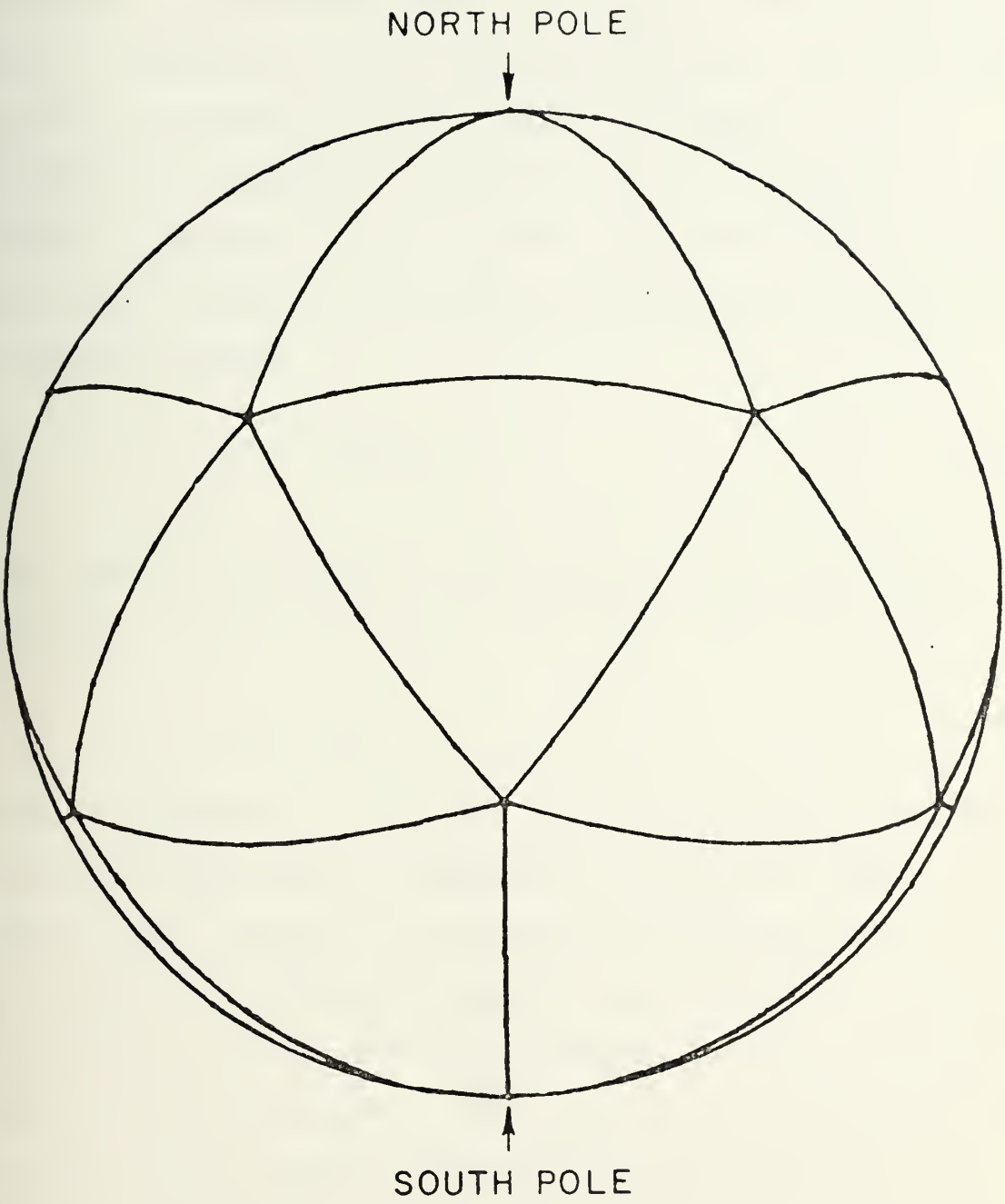

FIGURE 4. Regular icosahedron

The icosahedral grid used by this model consisted of a subdivision of each of the twenty major spherical triangles. Each major spherical triangle side was divided into n segments and the points were joined by arcs of great circles to produce a grid as shown in Figure 5. The mathematics are shown in greater detail in Appendix B. The number of points, N, in the grid where each side of a major spherical triangle is divided into n pieces is given by the following formula

$$N = 10n^2 + 2 \qquad \text{(III-4)}$$

The number of triangles, T, generated by the same grid is given by

$$T = 20n^2 \qquad \text{(III-5)}$$

With the exception of the vertices of the major spherical triangles, each node was supported by six other nodes. At the twelve major vertices, the nodes were supported by five nodes. The global correspondence number given each node was assigned by starting at the north pole and going around the latitude band as shown in Figure 6. The points of the triangles lying on zero and 360 degrees longitude were assigned the same global correspondence number to maintain cyclic continuity. This can be seen in Figure 7.

3. Global Correspondence Table

A Galerkin finite element approach requires inner products of the dependent variables, say f, with a pyramid

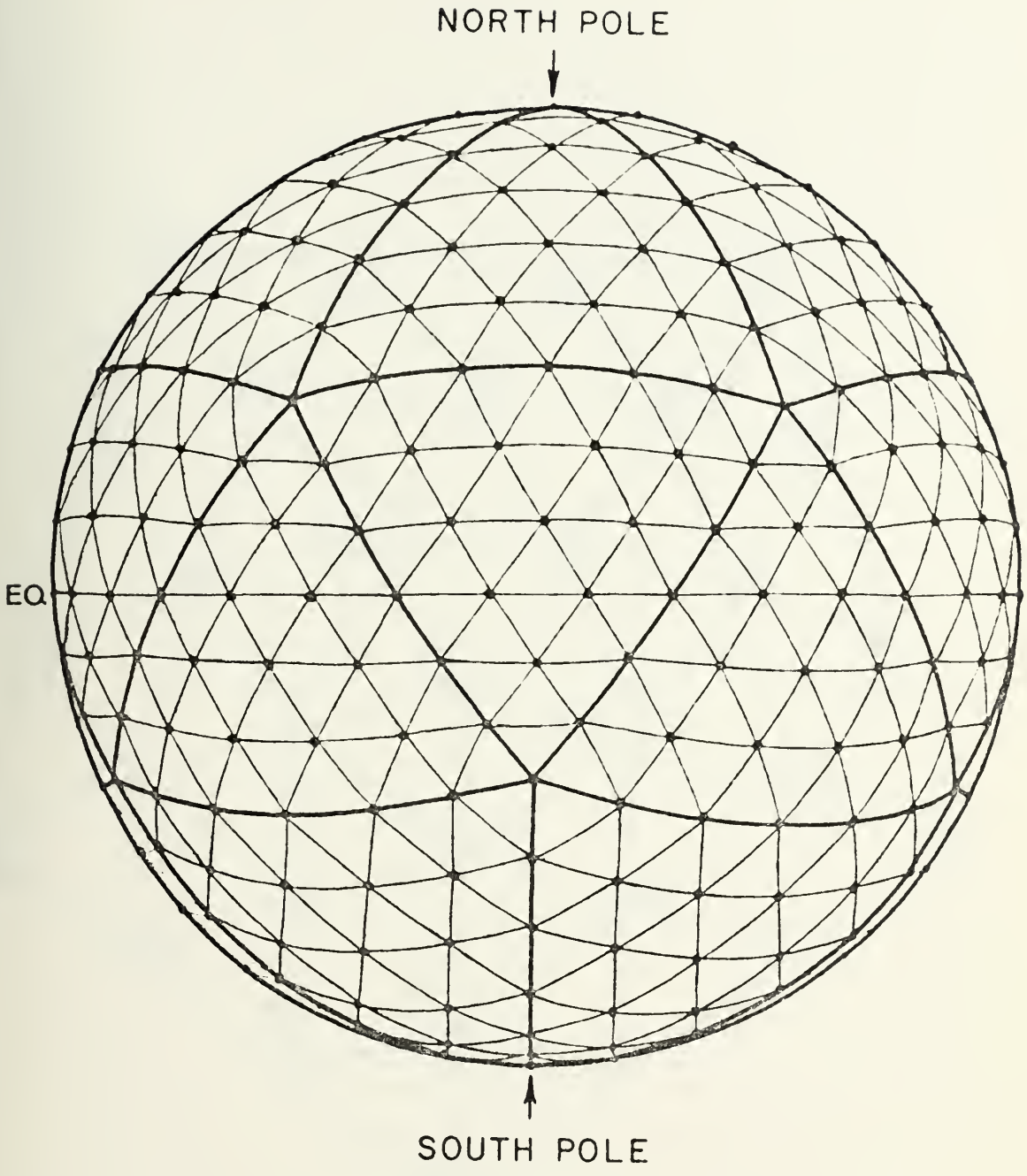

FIGURE 5. Icosahedral grid

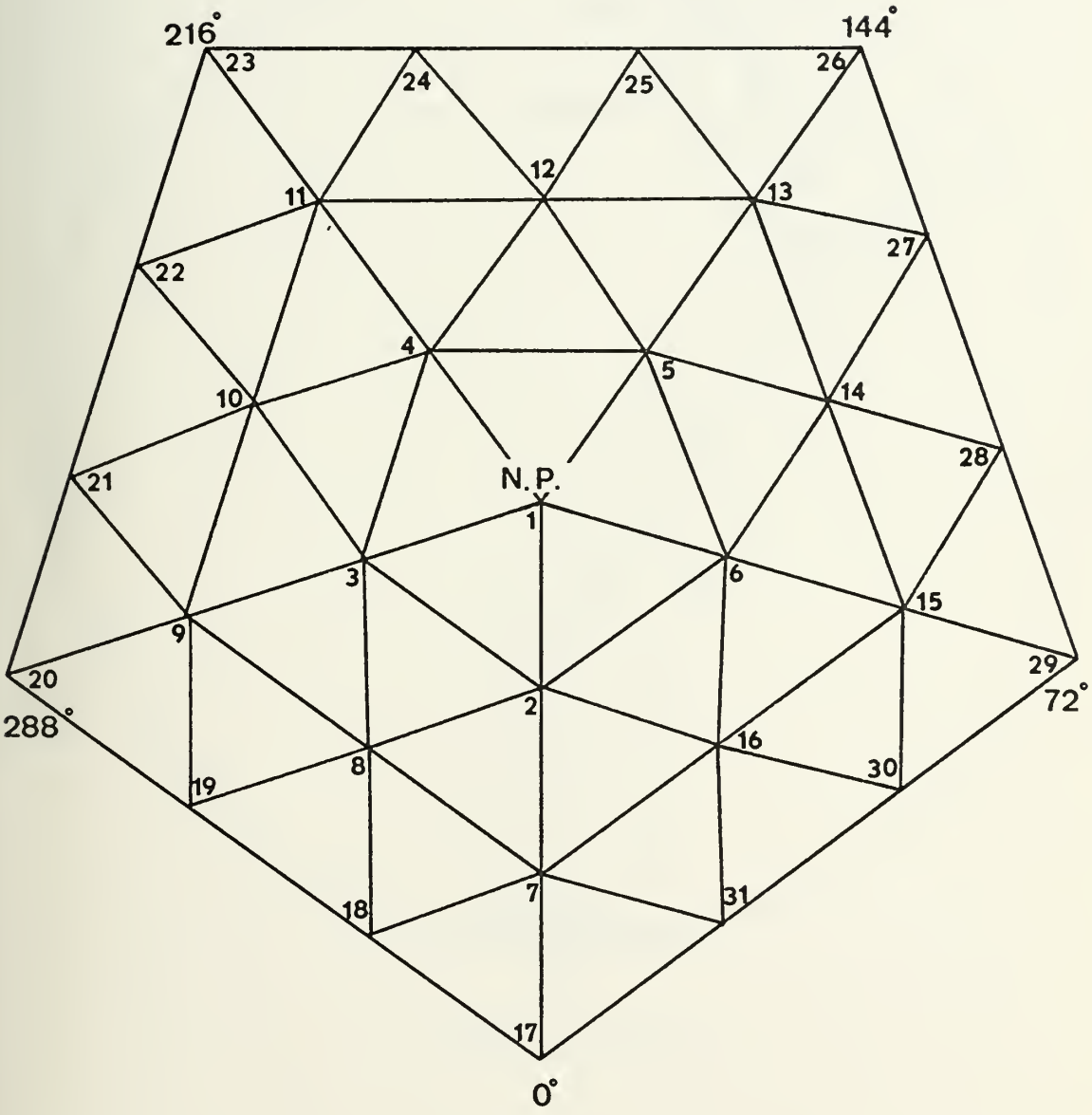

FIGURE 6. Global correspondence numbering scheme

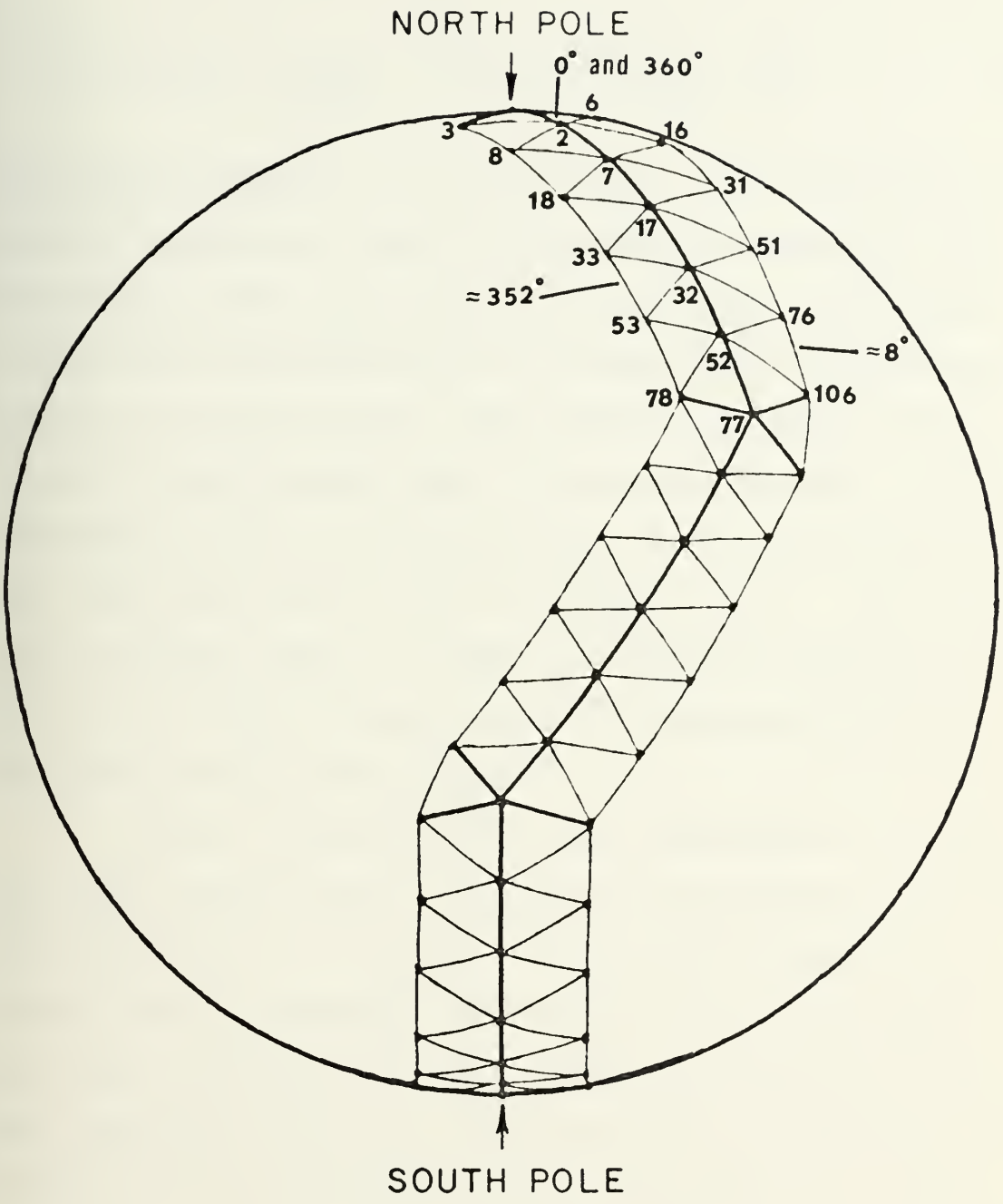

FIGURE 7. Elements displaying cyclic continuity

function, V_i. The inner product in spherical coordinates is defined as

$$\langle f(\lambda,\theta), V_i \rangle \equiv \iint_{\text{global}} f(\lambda,\theta)\, V_i\, a^2 \cos\theta \, d\lambda d\theta \qquad \text{(III-6)}$$

The pyramid function V_i, at any point i, has value one at i, linearly decreases to zero at the surrounding points and remains zero over the remainder of the globe. The global integration can be performed by integrating the particular function on each triangle, and placing the value of the integral in the proper place in the global matrix. A further explanation of the interaction between basis functions and test functions can be found in Section III.C.1. The global correspondence table is a means of properly scattering each triangle's surface integral into the appropriate space in the global coefficient matrix.

Given a triangle with local coordinates 1, 2, and 3, the inner product over the triangle will produce a value for the three points 1, 2, and 3. Point 1 will receive the values from the interplay with itself (1,1), with point 2 (1,2) and with point 3 (1,3). Point 2 will receive values from interplay with (2,1), (2,2) and (2,3). Similarly point 3 will receive values from (3,1), (3,2) and (3,3). Each triangle has a local 3 x 3 matrix which needs to be scattered into its global position in the coefficient matrix. Since each of the N points are multiplied by N global pyramid

functions, an N x N global coefficient matrix is formed during a complete integration over the globe for all points. Each row, i, in the matrix represents the equation for point i. A table had to be set up which correlates the local 3 x 3 matrix to its place in the global N x N matrix. The first step in developing a global correspondence table is to number all the nodes. The numbering scheme should be chosen so as to minimize the bandwidth of the coefficient matrix. For this paper, another approach was taken which reduced the storage requirement to a minimum (Section III.C.3). George (1971) includes a subroutine which, using any grid, will optimumly arrange the correspondence table to provide minimum bandwidth.

The icosahedral grid is indexed by starting at the north pole, point 1, and numbering around the next consecutive latitude band until all latitude bands are completed. The next step is to impose a local 1-2-3 coordinate onto each triangle and compile a local versus global table. Table I shows the first 10 triangles and its global correspondence table. The correspondence table is kept in a matrix and called during area integrations to scatter the 3 x 3 local matrix into the proper place in the coefficient matrix. The Computer Program section contains a program to generate the correspondence table.

TRIANGLE NUMBER	LOCAL COORDINATE		
	1	2	3
1	1	2	3
2	1	3	4
3	1	4	5
4	1	5	6
5	1	6	2
6	2	7	8
7	2	8	3
8	3	8	9
9	3	9	10
10	3	10	4

TABLE I. Global correspondence numbers for the first ten triangles

C. FINITE ELEMENT APPLICATION

1. Area Integrations

The inner product $\langle V_j, V_i \rangle$ can be computed by performing the necessary integration over each triangle separately and distributing the values to the proper position in the matrix. The actual integration can be done by a numerical scheme such as Gauss quadrature, or by use of an analytic approach. Due to the low order polynomials (linear) used, exact expressions may be derived for the integration of any function over an element. These formulas are used through a coordinate system called area coordinates. Strang and Fix (1973) state that area coordinates are known to engineers as triangular coordinates and to mathematicians as barycentric coordinates. Desai and Abel (1972) call them natural coordinates.

In the formulation, it becomes necessary to perform the inner product $\langle V_j, V_i \rangle$. Given the three pairs of coordinates (x,y), of the triangle in the x,y plane, there are three corresponding coordinates (L_1, L_2, L_3) in the natural coordinate system. This is shown in Figure 8. Zienkiewicz (1971) shows the relationship between a point x,y and the area coordinates (L_1, L_2, L_3), as

$$x = L_1 x_1 + L_2 x_2 + L_3 x_3 \qquad \text{(III-7)}$$

$$y = L_1 y_1 + L_2 y_2 + L_3 y_3 \qquad \text{(III-8)}$$

$$1 = L_1 + L_2 + L_3 \qquad \text{(III-9)}$$

where $L_1 = A_1/A$, $L_2 = A_2/A$, $L_3 = A_3/A$

and A = total area of triangle; and A_1, A_2, A_3 = area of the smaller triangles. He also shows that

$$\iint_A L_1^{\,a} L_2^{\,b} L_3^{\,c} \, dxdy = \frac{a!\, b!\, c!}{(a+b+c+2)!} 2A \quad . \qquad \text{(III-10)}$$

This formula was used to perform the integration. For example, the inner product $\langle V_j, V_i \rangle$ at j=i=1 is

$$\langle V_1, V_1 \rangle = \iint_A V_1^{\,2} \, dxdy = \frac{2!\, 0!\, 0!}{(2+0+0+2)!} 2A$$

$$= \frac{2}{24} \cdot 2A = \frac{A}{6}$$

and the inner product $\langle V_j, V_i \rangle$ at point j=2, i=1 is

$$\langle V_2, V_1 \rangle = \iint_A V_2 V_1 \, dxdy = \frac{1!\, 1!\, 0!}{(1+1+0+2)!} 2A$$

$$= \frac{1}{24} \cdot 2A = \frac{A}{12} \quad .$$

With this formula, any necessary product of basis functions can be integrated over one triangle.

If equations (III-7), (III-8), and (III-9) are now solved for L_1, L_2, L_3 and written in matrix form the result is

$$\begin{bmatrix} L_1 \\ L_2 \\ L_3 \end{bmatrix} = \frac{1}{2A} \begin{bmatrix} 2A & b_1 & a_1 \\ 2A & b_2 & a_2 \\ 2A & b_3 & a_3 \end{bmatrix} \begin{bmatrix} 1 \\ x \\ y \end{bmatrix} \qquad \text{(III-11)}$$

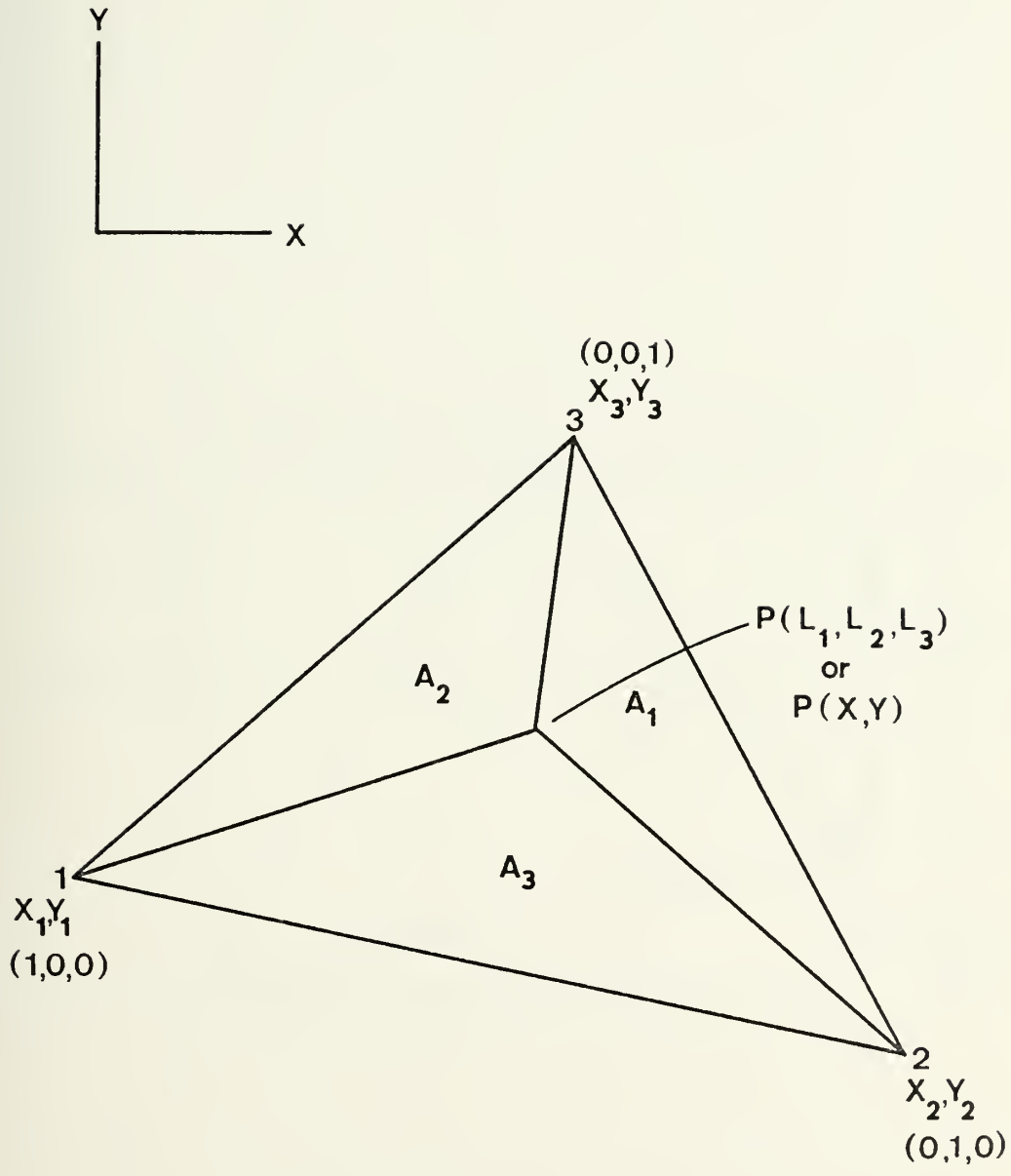

FIGURE 8. Cartesian coordinates vs. natural coordinates

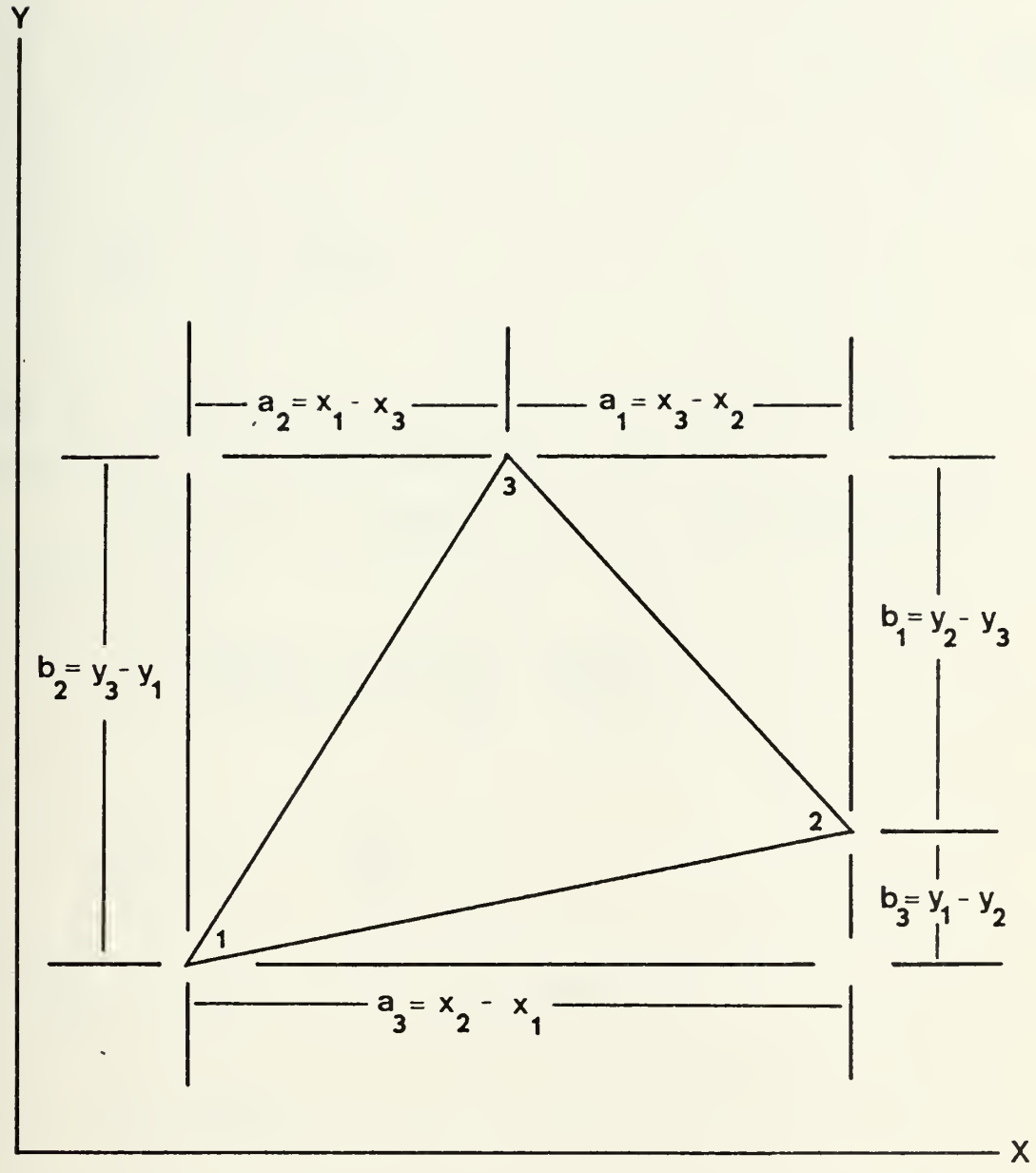

FIGURE 9. Triangle definitions for area coordinates

as shown in Desai and Abel (1972) where the a's and b's are as defined in Figure 9. Differentiation of (III-11) shows that

$$\frac{\partial}{\partial x} = \sum_{i=1}^{3} \frac{b_i}{2A} \frac{\partial}{\partial L_i} \qquad \text{(III-12)}$$

$$\frac{\partial}{\partial y} = \sum_{i=1}^{3} \frac{a_i}{2A} \frac{\partial}{\partial L_i} \qquad \text{(III-13)}$$

These two equations will be used when evaluating the derivatives of the basis and test functions.

For example, assuming $V_j = L_j$, the derivative V_{jx} at point 1 is

$$V_{1x} = \frac{b_1}{2A} \frac{\partial V_1}{\partial L_1} + \frac{b_2}{2A} \frac{\partial V_1}{\partial L_2} + \frac{b_3}{2A} \frac{\partial V_1}{\partial L_3}$$

but

$$\frac{\partial V_1}{\partial L_2} = 0 \quad , \quad \frac{\partial V_1}{\partial L_3} = 0$$

and $\frac{\partial V_1}{\partial L_1} = 1$ at point 1. Consequently,

$$V_{1x} = \frac{b_1}{2A} .$$

The inner product $\langle V_{jx}, V_i \rangle$ at point j=2, i=1 becomes

$$\langle V_{2x}, V_1 \rangle = \iint_A \frac{b_2}{2A} V_1 \, dxdy$$

$$= \frac{b_2}{2A} \frac{1!\ 0!\ 0!}{(1+0+0+2)!} 2A$$

$$= \frac{b_2}{6} .$$

Thus with these simple formulas, all the inner products can readily be evaluated.

2. Nonlinear Terms

The nonlinear terms in the system of equations were quasi-linearized by the time extrapolated Crank-Nicholson method. When a uu_λ term was encountered, it was replaced by u^*u_λ where

$$u^* = u^{N+\frac{1}{2}} \approx \frac{3}{2} u^N - \frac{1}{2} u^{N-1} \qquad \text{(III-14)}$$

Thus when solving for time level N+1, all the * quantities will be known. In this manner the nonlinear terms are quasi-lienarized in time to facilitate integration. The inner product $\langle \alpha_j \alpha_k^* V_k V_{j\lambda}, V_i \rangle$ represents u^*u_λ in the notation of this paper. In matrix notation this becomes

$$\alpha_j A_{ij} = \alpha_j \langle \alpha_k^* V_k V_{j\lambda}, V_i \rangle \qquad \text{(III-15)}$$

In the local 3 x 3 matrix, any point within the local matrix becomes

$$A_{ij} = \langle \alpha_1^* V_1 V_{j\lambda}, V_i \rangle + \langle \alpha_2^* V_2 V_{j\lambda}, V_i \rangle + \langle \alpha_3^* V_3 V_{j\lambda}, V_i \rangle . \qquad \text{(III-16)}$$

In this manner the known * functions are integrated into the Galerkin space. The nonlinear terms are "linearized." Each equation becomes one equation with one unknown. The three equations are coupled when the new information at time level

N+1 is used to solve the equations at time level N+2. A constraint on a linearization scheme is that a large time step cannot be taken. The time step should not be so large as to cause the change in the variable to be larger than the truncation error of the approximation. This linearization, which uncouples the three equations, causes some inconsistency with respect to the three dependent variables.

3. Matrix Storage

During a global integration, an N x N coefficient matrix is generated which is very sparse due to the fact that the maximum number of triangles supporting any one point is six. These six triangles provide interaction between the seven points involved. Each row, i, in the N x N matrix therefore represents the equations written down at point i, and each row would have, at a maximum, seven entries. If the matrix were condensed by omitting all terms identically zero, the size would be N x 7. This represents a sizable reduction in core storage.

The method of condensed matrix storage offers the advantage of minimum core storage, For large fields, a complete N x N matrix can easily exceed core capabilities of most computers. The storage of a sparse matrix would be extremely wasteful, hence, the condensed method was adopted. If the coefficient matrix were symmetric or had a narrow bandwidth, then storage could be easily accomplished using a smaller amount of computer core. The icosahedral grid offers neither a symmetric matrix nor one with a narrow

bandwidth. The disadvantage of the matrix storage scheme was the need to search a correlation matrix when a value needed to be placed in the coefficient matrix. An efficient searching routine was developed and can be seen in the Computer Program section under Subroutine SEARCH. But, unfortunately, this subroutine had to be executed many times during the time integrations.

In this model, an iterative procedure was used to solve the coefficient matrix. In order to reduce the N x N matrix to an N x 7 matrix, a correlation table had to be developed. The correlation table, a separate N x 7 matrix, was assembled which contained the seven points involved in any one row of the coefficient matrix. An example will show more clearly the process involved. Table II and Figure 10 show the triangle, points and correlation table involved with point 2. In an N x N matrix, point 2's equation (row 2) would have non-zero entries in position (2,1), (2,2), (2,3), (2,6), (2,7), (2,8), and (2,16). For example, in the N x 7 coefficient matrix, entry (2,16) would be stored in (2,7) of the condensed matrix and entry (2,8) would be stored in (2,6). During matrix multiplication, the proper address of a vector component had to be selected as well as the proper address in the N x 7 matrix. This can be accomplished by calling, for example, position (2,7) of the correlation matrix to find that entry (2,7) of the coefficient matrix goes with entry 16 of the vector. The process is reversed when it

Legend (1) Triangle number 1

1 Point number 1

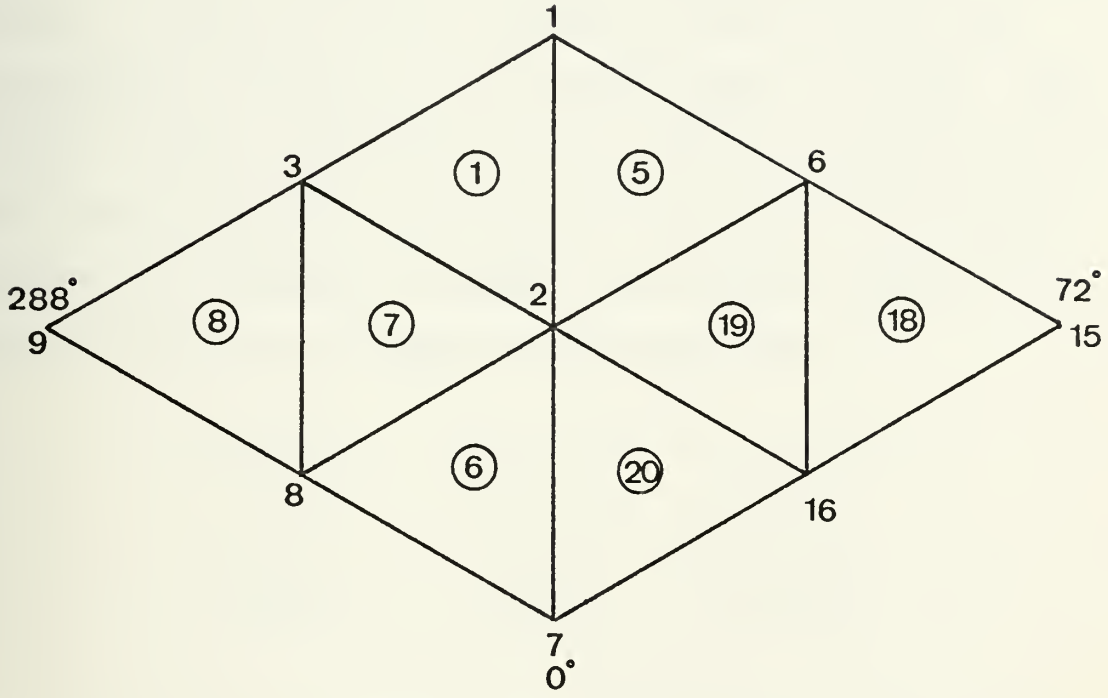

Figure 10. Diagram of triangles 1, 5, 6, 7, 19, 20 and nodes 1, 2, 3, 6, 7, 8, 16.

TABLE II. Correlation table for nodes 2, 7, 8

CORRELATION TABLE

N x 7 Matrix

Row	Column						
	1	2	3	4	5	6	7
2	1	2	3	6	7	8	16
7	2	7	8	16	17	18	31
8	2	3	7	8	9	18	19
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
N	–	–	–	–	–	–	–

becomes necessary to scatter the local 3 x 3 matrix built during each triangle's surface integration. The global number of each point in the triangle determines which row into the N x 7 coefficient matrix the value will go. The column is found by doing a search of the row from the correlation matrix until the second number is found. The column of the correlation table in which the second number was found determines the column in the N x 7 coefficient matrix. For example, triangle 6 will have a local 3 x 3 matrix with global numbers which looks like

$$\begin{bmatrix} (2,2) & (2,7) & (2,8) \\ (7,2) & (7,7) & (7,8) \\ (8,2) & (8,7) & (8,8) \end{bmatrix}$$

Row 2 of the coefficient matrix will contain the values of (2,2), (2,7), and (2,8). A search of correlation matrix's row 2 shows that (2,8) of the N x N matrix goes into (2,6) of the N x 7 matrix.

The correlation matrix is developed by searching the global correspondence table for the six triangles that contain point i. These six triangles are then compared and sorted to produce the seven points interacting with point i. The seven points are then arranged in ascending order.

A copy of the program which produced the correlation table can be found in the Computer Program section.

IV. INITIAL CONDITIONS

The initial conditions used in this paper were computed from an analytic solution to the nonlinear balance equation.

Haurwitz (1940) developed a stream function and showed that harmonic waves computed from this stream function will move with constant angular velocity in a non-divergent barotropic atmosphere. The stream function, ψ, used by Haurwitz is given by

$$\psi = A^* \sin(m\lambda - \nu t) \sin\theta \cos m\theta - Ba^2 \sin\theta \qquad \text{(IV-1)}$$

where A^* and B are constants, m is the wave number, ν is the angular wave velocity, a is the radius of the earth.

The constant B is related to the wave number and angular wave velocity by

$$\frac{\nu}{m} = \frac{B\,M(M+1) - 2}{M(M+1)} - \frac{2\Omega}{M(M+1)} \qquad \text{(IV-2)}$$

where (ν/m) is the angular phase speed, Ω is the angular velocity of the earth and $M = m+1$. To allow a phase speed propagation comparison with a finite difference scheme, the constant A^* was arbitrarily chosen to be the same as the A in a M.S. thesis by Monaco (1975). Phillips (1959) used the Haurwitz stream function, ψ, in the nonlinear balance equation to determine ϕ. The nonlinear balance equation is

$$\nabla^2\phi = f\nabla^2\psi - \nabla\psi\cdot\nabla f - 2\left(\frac{\partial^2\psi}{\partial x\partial y}\right)^2 + 2\left(\frac{\partial^2\psi}{\partial x^2}\frac{\partial^2\psi}{\partial y^2}\right) \qquad \text{(IV-3)}$$

which in spherical coordinates becomes

$$\frac{1}{a^2}\left[\frac{1}{\cos^2\theta}\frac{\partial^2\phi}{\partial\lambda^2} + \frac{1}{\cos\theta}\frac{\partial}{\partial\theta}\left(\cos\theta\frac{\partial\phi}{\partial\theta}\right)\right]$$

$$= \frac{f}{a^2}\left[\frac{1}{\cos^2\theta}\frac{\partial^2\psi}{\partial\lambda^2} + \frac{1}{\cos\theta}\frac{\partial}{\partial\theta}\left(\cos\theta\frac{\partial\psi}{\partial\theta}\right)\right]$$

$$+ \frac{1}{a^2}\frac{\partial\psi}{\partial\theta}\frac{\partial f}{\partial\theta} - 2\left(\frac{1}{a^2\cos\theta}\frac{\partial^2\psi}{\partial\lambda\partial\theta}\right)^2$$

$$+ \frac{2}{a^4\cos^2\theta}\frac{\partial^2\psi}{\partial\lambda^2}\frac{\partial^2\psi}{\partial\theta^2} \qquad \text{(IV-4)}$$

Phillips (1959) found the solution to the nonlinear balance equation for the geopotential distribution, ϕ, to be

$$\phi = a^2 A(\theta) + a^2 B(\theta)\sin(m\lambda) + a^2 C(\theta)(2\sin^2 m\lambda - 1) \qquad \text{(IV-5)}$$

where

$$A(\theta) = \frac{B}{2}(2\Omega+B)\cos^2\theta + \frac{1}{4}\left(\frac{A^*}{a^2}\right)^2\cos^{2m}\theta\left[(m+1)\cos^2\theta + (2m^2-m-2) - \frac{2m^2}{\cos^2\theta}\right] \qquad \text{(IV-5.1)}$$

$$B(\theta) = \frac{2(\Omega+B)\frac{A^*}{a^2}}{(m+1)(m+2)}\cos^m\theta\left[(m^2+2m+2) - (m+1)^2\cos^2\theta\right] \qquad \text{(IV-5.2)}$$

and $$C(\theta) = \frac{1}{4}\left(\frac{A^*}{a^2}\right)^2 \cos^{2m}\theta \left[(m+1)\cos^2\theta - (m+2)\right] \quad \text{(IV-5.3)}$$

The foregoing geopotential disturbance, ϕ, must be added to the mean height for the level desired. The mean height was obtained from the NACA standard atmosphere (Haltiner and Martin, 1957). Although an attempt has been made to apply the model at the level of non-divergence, the equations used in the model permit divergence. Phillips (1959) found that the presence of divergence will cause the waves in the height fields to move slightly slower than a non-divergent model, especially for small wave numbers.

Analytic initial winds can be obtained from the stream function, ψ, as follows

$$u = -\frac{1}{a}\frac{\partial\psi}{\partial\theta} \quad \text{(IV-6)}$$

and

$$v = \frac{1}{a\cos\theta}\frac{\partial\psi}{\partial\lambda} \quad \text{(IV-7)}$$

From ψ, in IV-1, the values of u and v are

$$u = -\frac{1}{a}\left[A^*\sin(m\lambda-\nu t)\cos^{m+1}\theta - mA^*\sin(m\lambda-\nu t)\cos^{m-1}\theta\sin^2\theta - Ba^2\cos\theta\right] \quad \text{(IV-8)}$$

$$v = \frac{1}{a}\left[A^*m\sin\theta\cos^{m-1}\theta\cos(m\lambda-\nu t)\right] \quad \text{(IV-9)}$$

These analytically determined winds from the nonlinear balance equation were chosen because they are in approximate gradient balance and hence more realistic than simply geostrophic winds. Subroutine SOLUT of the main program, Computer Program section, produced the initial fields for the latitude and longitude of each point.

V. WAVE ANALYSIS METHOD

A harmonic analysis was performed around constant latitude circles to determine the phase speed and amplitude of meteorological waves. The initial fields were analyzed first to provide a reference condition.

The Fourier series used was

$$F(x) = A_o + \sum_m (A_m \cos mx + B_m \sin mx) \qquad (V\text{-}1)$$

which can be represented as one trigonometric function given by

$$F(x) = C_o + \sum_m C_m \cos(mx - \delta_m) \qquad (V\text{-}2)$$

where

$$C_m = \frac{B_m}{\sin \delta_m} = \frac{A_m}{\cos \delta_m} \qquad (V\text{-}3)$$

and

$$\delta_m = \tan^{-1} \frac{B_m}{A_m} \qquad (V\text{-}4)$$

In order to analyze phase speed propagation, nonlinear instability and energy conservation, numerous wave numbers, amplitudes and phase speeds were examined.

VI. GRID CONVERSIONS

The grid that is most compatible to this finite element model is the icosahedral grid. But this grid is incompatible with the grids needed to perform harmonic analysis or graphical display. This fact could be a serious drawback to an operational finite element model. An advantageous characteristic of the finite element method is that the solutions are continuous over the element. For this model, linear functions were used. To convert the icosahedral solutions to solutions at different points involves evaluating the function over the element at the point desired. This is where the finite element method will have an advantage over the finite difference schemes. Finite difference schemes must interpolate from discrete point values to an interior point. In this finite element method, the basis functions are of the form

$$f = d + b\lambda + e\theta \tag{VI-1}$$

The value of the funciton is known at three points. These three equations can be written for points 1, 2, and 3

$$f_1 = d + b\lambda_1 + e\theta_1 \tag{VI-2}$$

$$f_2 = d + b\lambda_2 + e\theta_2 \tag{VI-3}$$

$$f_3 = d + b\lambda_3 + e\theta_3 \tag{VI-4}$$

We have a system of three equations and three unknowns. The value of the coefficients d, b, and e over any one element can be easily found. Once the coefficients are known, the value of the function at any point λ,θ in the element can be found by evaluating the polynomial given in equation (VI-1).

For this model a 72 x 35 regular 5° longitude-latitude grid was required for harmonic analysis and graphical output. A subroutine was developed which correlated each point in the 72 x 35 grid to its appropriate triangle in the icosahedral grid. This is subroutine SURVEY in the main program in the Computer Program section. The value of the coefficients d, b, and e in equation (VI-1) were found for each triangle by subroutine EVAL in the main program in the Computer Program section. The value of the function at any point in the 72 x 35 grid was evaluated by subroutine DISPL in the main program in the Computer Program section.

VII. EXPERIMENTS

Experiment 1. The model was run with the same conditions using the icosahedral grid and the λ,θ grid. The purpose of this experiment was to compare a varying Δx length grid, as most non projection finite difference models use, with a near constant Δx grid such as the icosahedral grid. The conditions used were a 10-minute time step, wave number 4, amplitude times wave number = $28. \times 10^7$ meters and a phase speed of 10° longitude/day.

Experiment 2. The model was run on the icosahedral grid with wave numbers 4, 8, and 12. The phase speeds were all 10° longitude/day. A 10-minute time step was used for all runs. The amplitude of the waves was changed so that the product of amplitude and wave number remained 28×10^7 meters. This will constrain the order of magnitude of the north-south component to remain the same. The purpose of this experiment was to observe phase speed propagation of the various waves and compare them to the phase speed propagation characteristics of various finite difference schemes.

Experiment 3. The model was run with increasing time steps with wave number 4. The other conditions were the same as in experiment number 1. Time steps of 10, 12, 15, and 18 minutes were run for a full 72 hour prognosis.

VIII. RESULTS

Experiment 1. Initially the model ran on the λ,θ grid but after twelve hours the harmonic analysis near the polar regions showed increasing amplitudes in the high wave numbers. Eventually the solutions in the polar regions caused the model to become unstable. There are several possible explanations for this instability. From a stability criterion viewpoint, the Δx near the poles was so small using the λ,θ grid that the 10-minute time step exceeded the stability cut off point. This is another major drawback of this grid. The decreasing Δx as the pole is approached is a common source of problems for normal finite difference schemes. From theoretical considerations, the finite element method should be well suited to an icosahedral grid and this should alleviate the converging Δx problem. The λ,θ grid has 36 points around each latitude circle. It can therefore resolve wave number 18. Any truncation error created while solving the matrix equations will appear as high frequency noise.

Another reason is that the equations are singular at the poles. In order to avoid these singularities, the model does not predict u, v or ϕ at the poles. Instead u and v are set to zero initially and are kept zero throughout the entire integration. A value for ϕ at the poles is found by averaging the first latitude circle down from the pole and then setting the pole and the next latitude circle equal to the average

value. This will flatten the cap over the pole and make the u and v assumptions at the poles more consistent, although obviously this treatment of the poles is not completely desirable.

Experiment 2. Figures 11, 12, and 13 show the results of the phase speed propagation test for wave numbers 4, 8, 12 respectively. While identical tests using finite difference schemes were not available, Maher (1974), in a Master's Thesis, performed phase speed propagation analysis with similar wave numbers and amplitudes using second and fourth order finite difference schemes with a comparable number of points. The comparison showed that the finite element method was more accurate than second or fourth order differencing for the cases examined.

Due to the coefficient chosen, the amplitude was minimal in the waves at higher latitude. On the right side of each figure is a value showing the number of points around each latitude band. As the number of points per each latitude band decreases, the phase speed propagation decreases. The finite element method therefore has the same relationship as finite differencing where phase propagation is concerned. It takes more points per wave to improve the phase propagation. Charts A-I show the wave propagation in 12 hour increments for the three wave numbers tested.

Experiment 3. With the three equations being treated separately at each time level and the main forcing functions, coriolis force, and pressure gradient force, being treated

explicitly, it is reasonable to expect a linear stability criterion approximately of the form

$$c \frac{\Delta t}{\Delta x} \leq .707$$

where c = the phase speed of the fastest waves and Δx the minimum Δx in the grid. The actual stability criterion for this model on the icosahedral grid has not rigorously been worked out. However, in notes by Archer (1975), the stability criterion for a simple wave equation was worked out and found to be comparable to that with similar finite difference schemes. The assumed stability criterion predicted a maximum time step of 19 minutes. It was found that the model with an 18 minute time step became unstable after 12 hours and after 24 hours with a 15 minute time step. However the model remained stable with a 12 minute time step up to 48 hours.

In all cases, instability in the equatorial regions was observed after a sufficient period of integration. For the wave number 4 case, this instability occurred after 84 hours of real time integration. The instability occurred sooner for larger time steps.

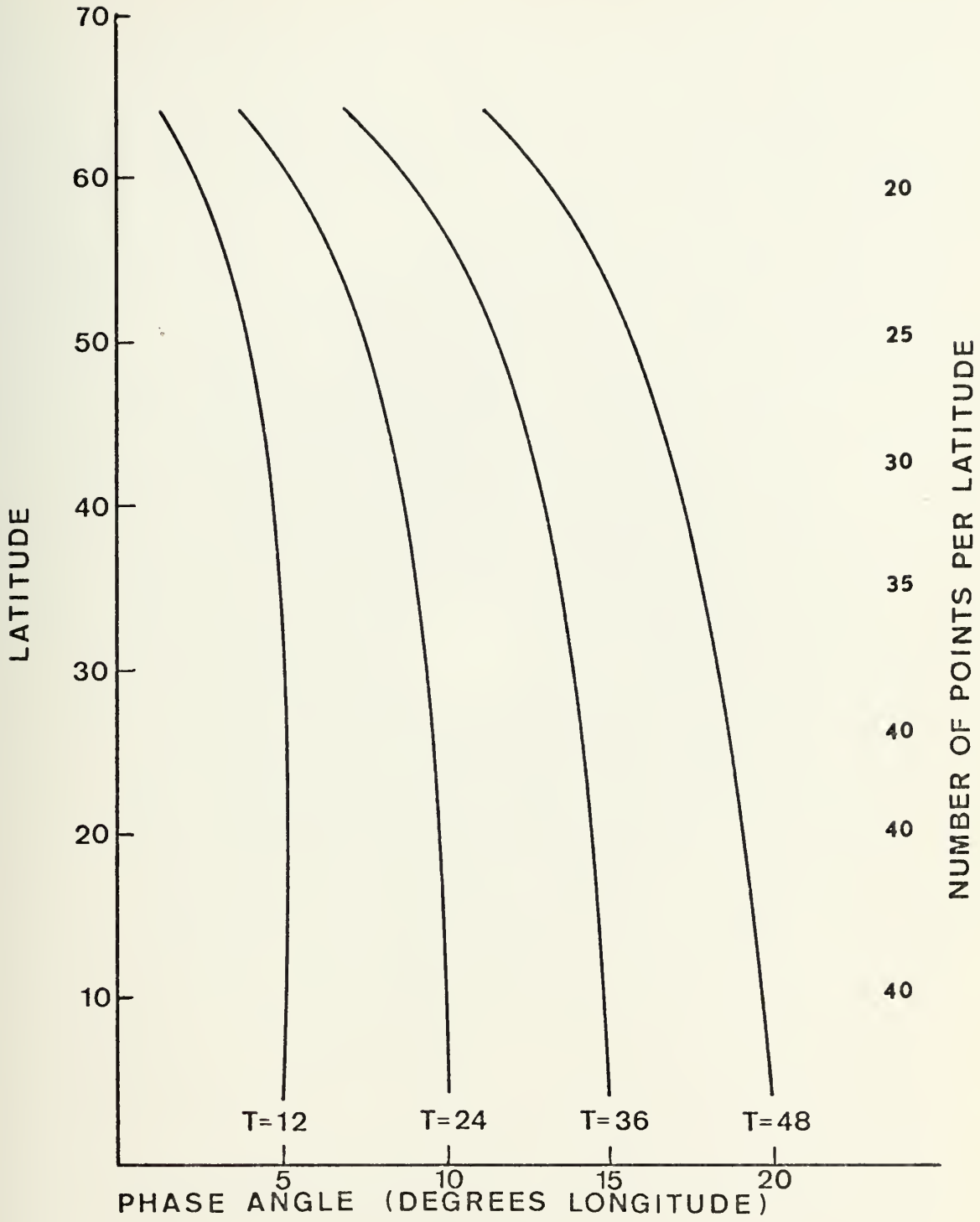

FIGURE 11. Phase angle (degrees longitude) vs. latitude for icosahedral grid, wave number 4, phase speed 10°/day and $A^* = 7.0 \times 10^7$. (Latitudes with near zero wave amplitude are not included and time is given in hours.)

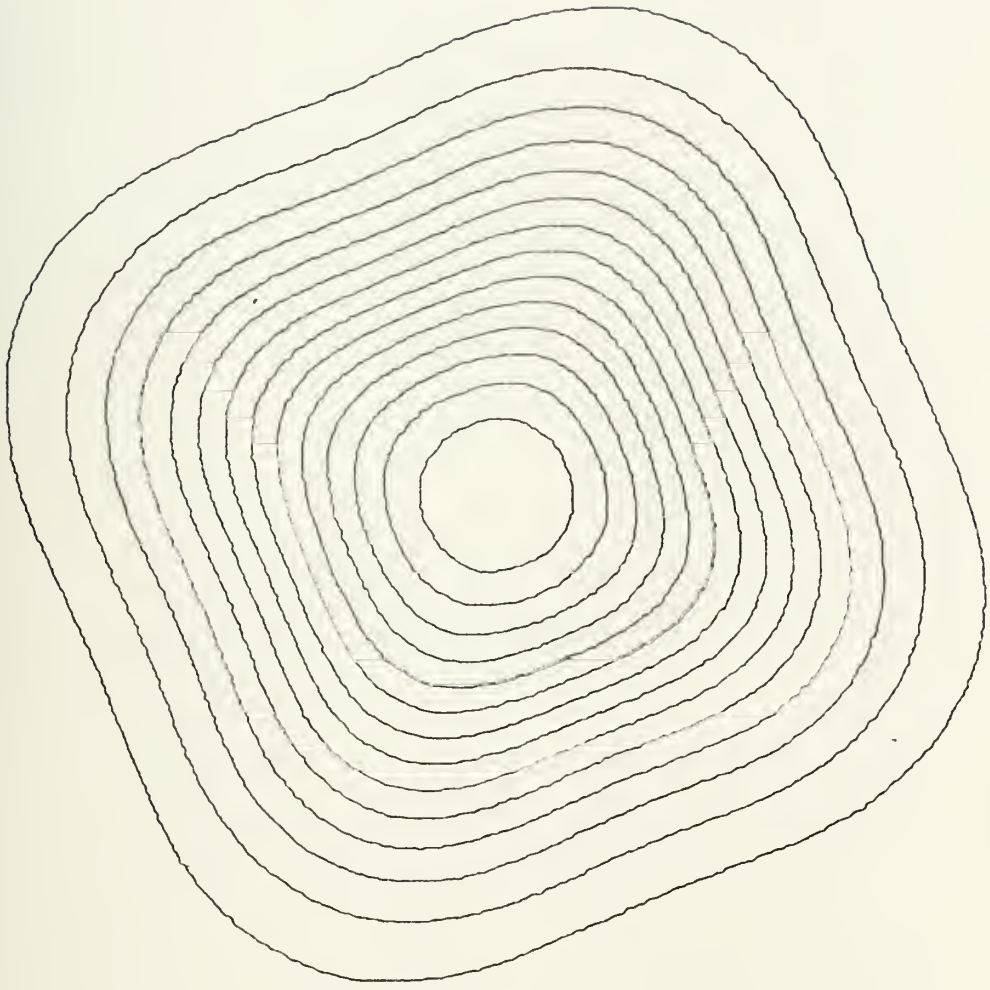

Chart A. Initial PHI field analysis, wave number 4, phase speed 10°/day, $A^* = 7.0 \times 10^7$.

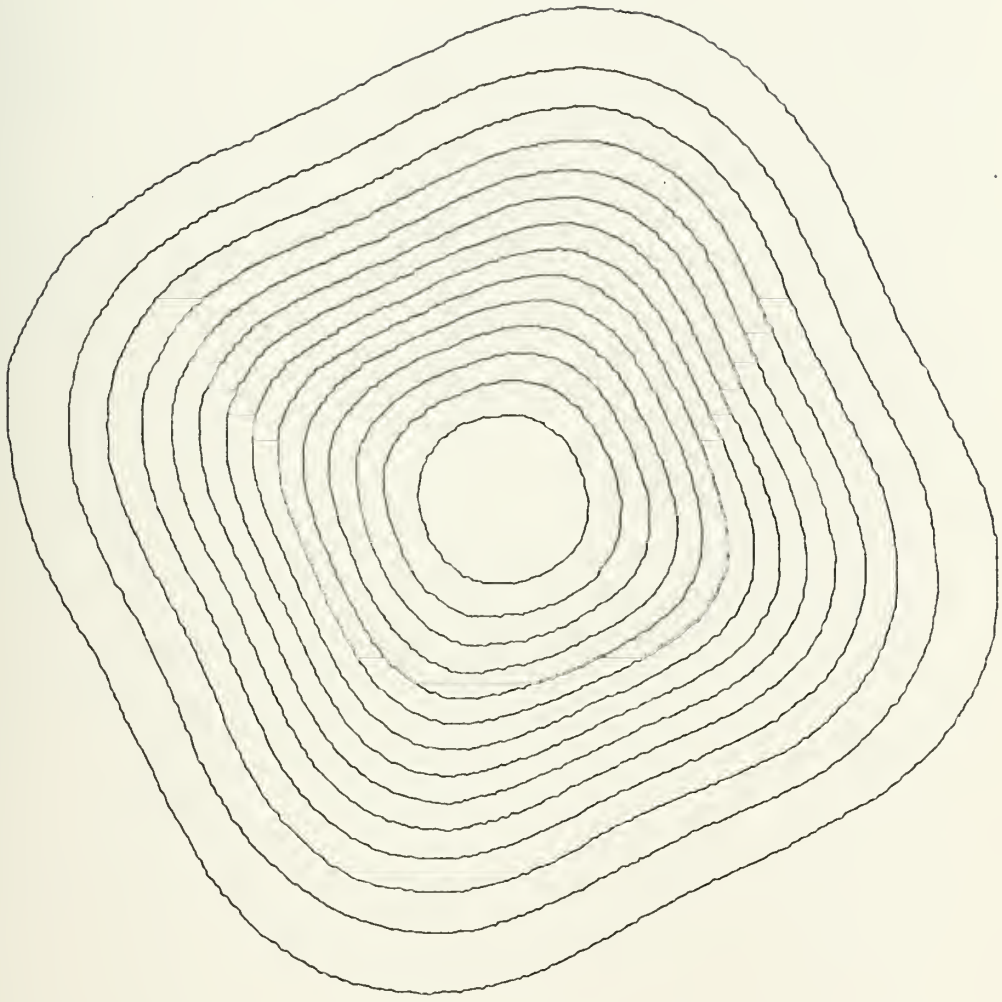

Chart B. 12-hour PHI field forecast, wave number 4, phase speed 10°/day, $A^* = 7.0 \times 10^7$.

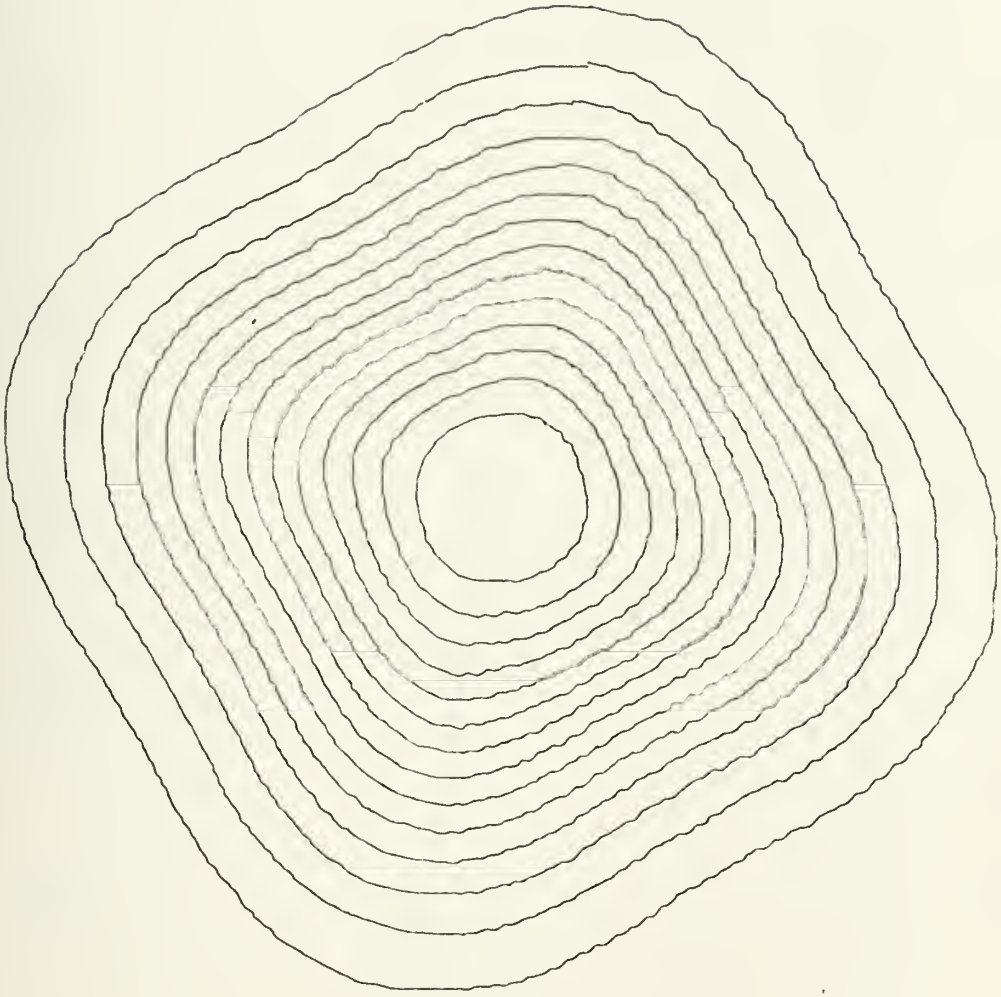

Chart C. 24-hour PHI field forecast, wave number 4, phase speed 10°/day, $A^* = 7.0 \times 10^7$.

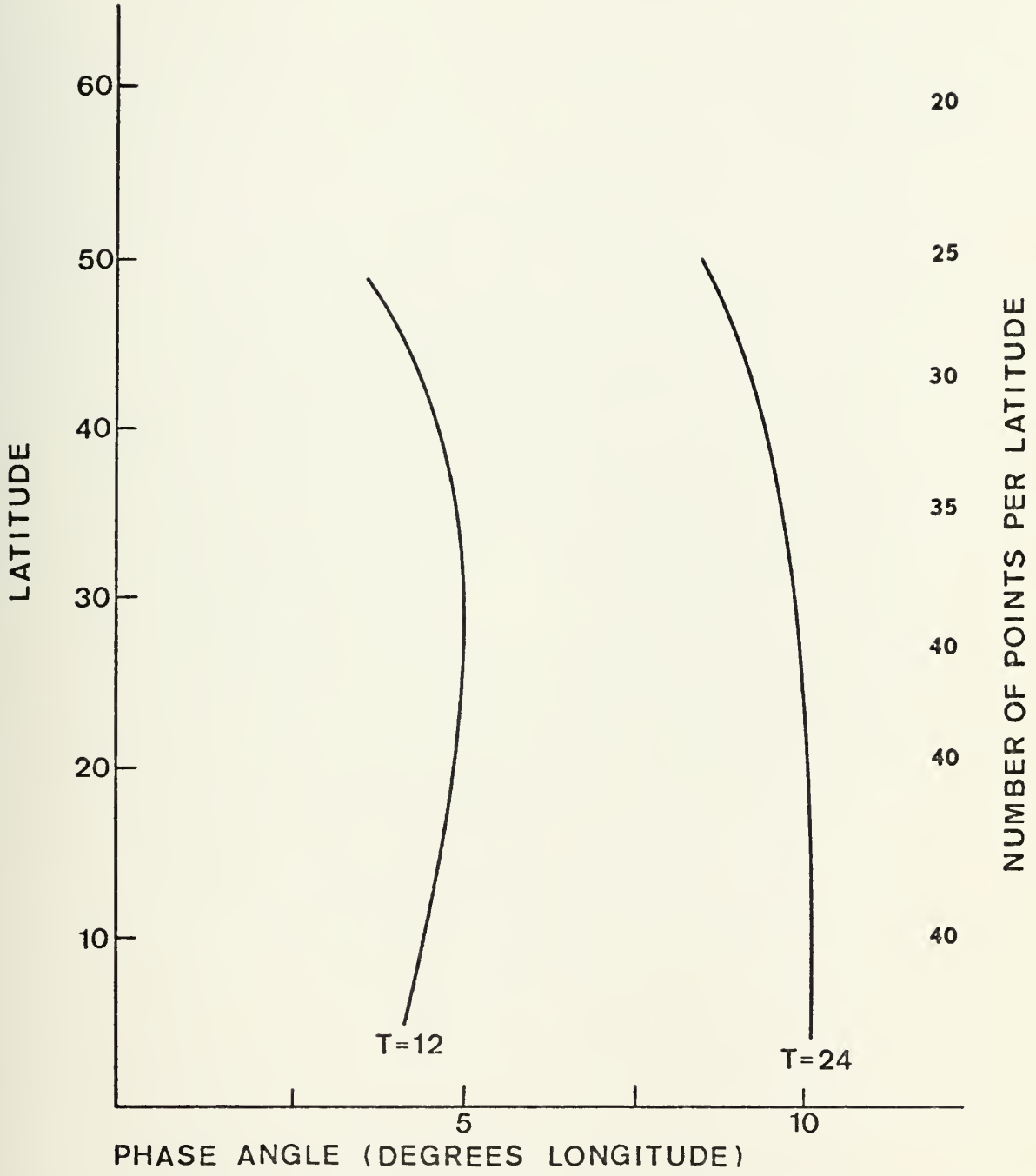

FIGURE 12. Phase angle (degrees longitude) vs. latitude for icosahedral grid, wave number 8, phase speed 10°/day and $A^* = 3.5 \times 10^7$. (Latitudes with near zero wave amplitude are not included and time is given in hours.)

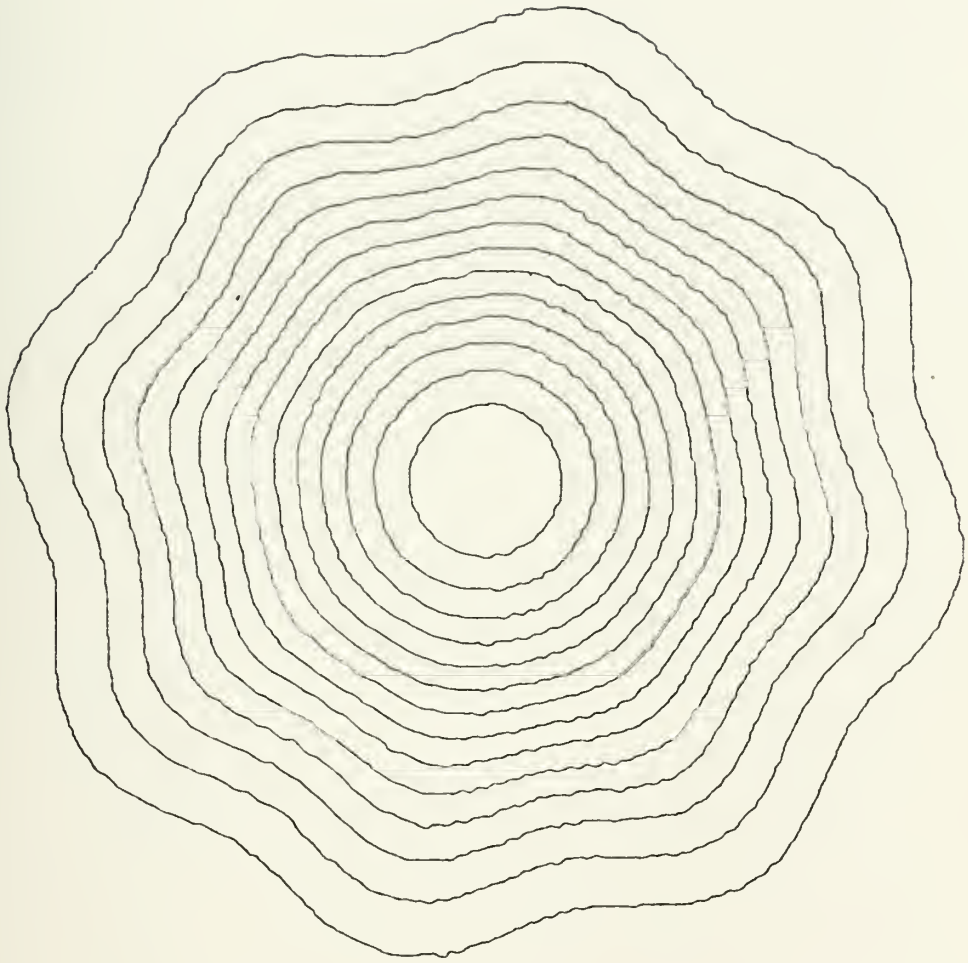

Chart D. Initial PHI field analysis, wave number 8, phase speed 10°/day, $A^* = 3.5 \times 10^7$.

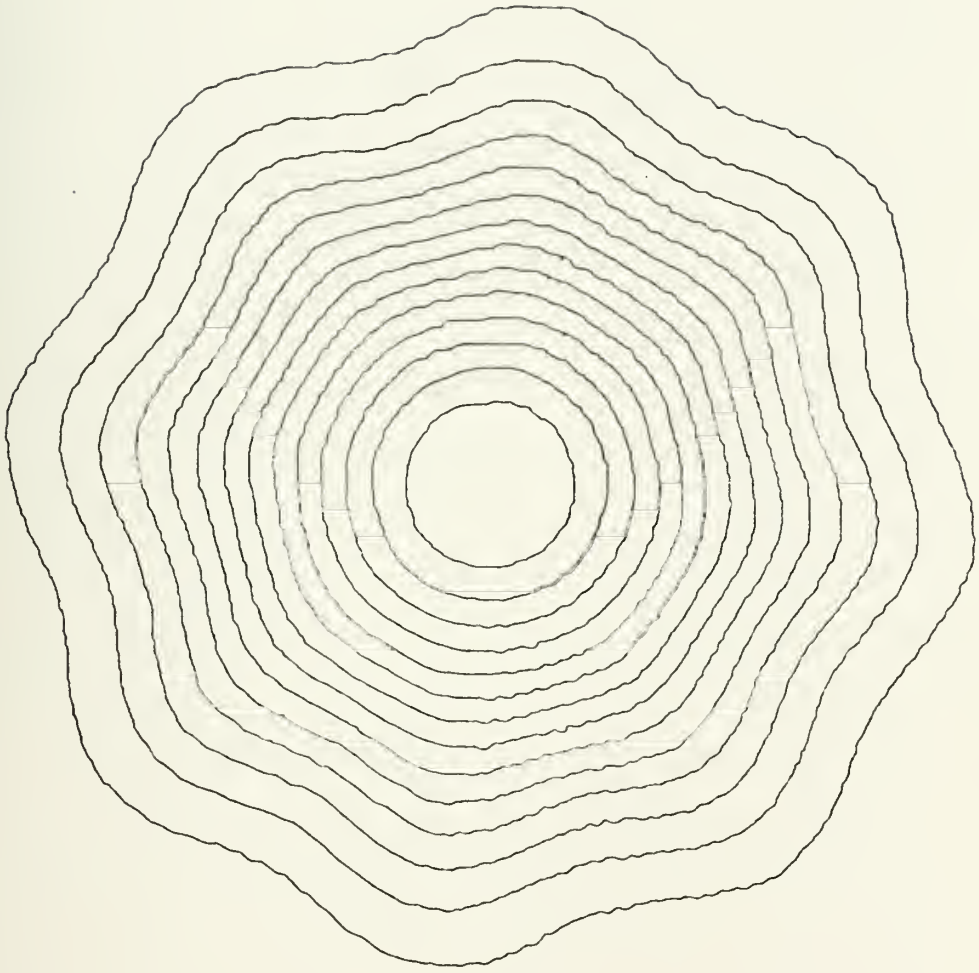

Chart E. 12-hour PHI field forecast, wave number 8, phase speed 10°/day, $A^* = 3.5 \times 10^7$.

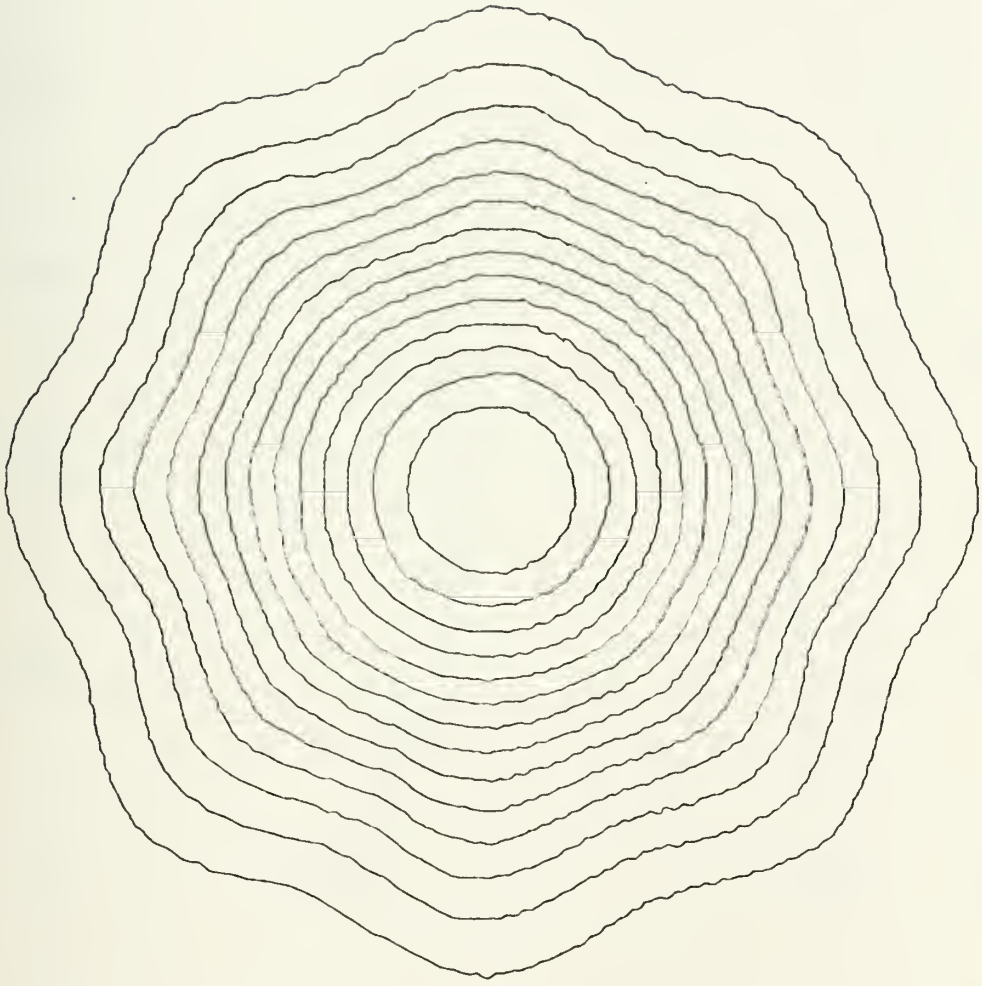

Chart F. 24-hour PHI field forecast, wave number 8, phase speed 10°/day, $A^* = 3.5 \times 10^7$.

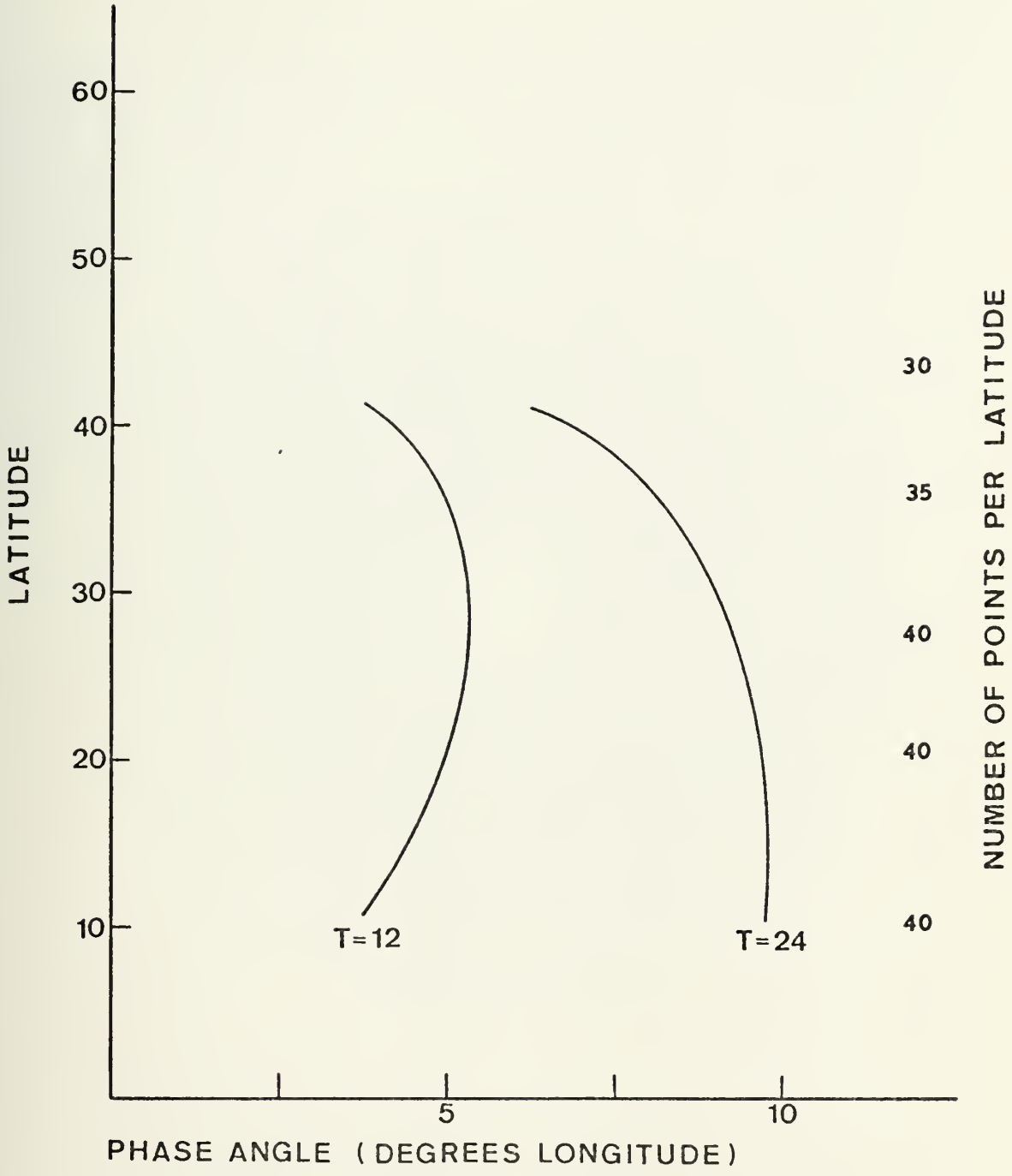

FIGURE 13. Phase angle (degrees longitude) vs. latitude for icosahedral grid, wave number 12, phase speed 10°/day and $A^* = 2.3 \times 10^7$. (Latitudes with near zero wave amplitude are not included and time is given in hours.)

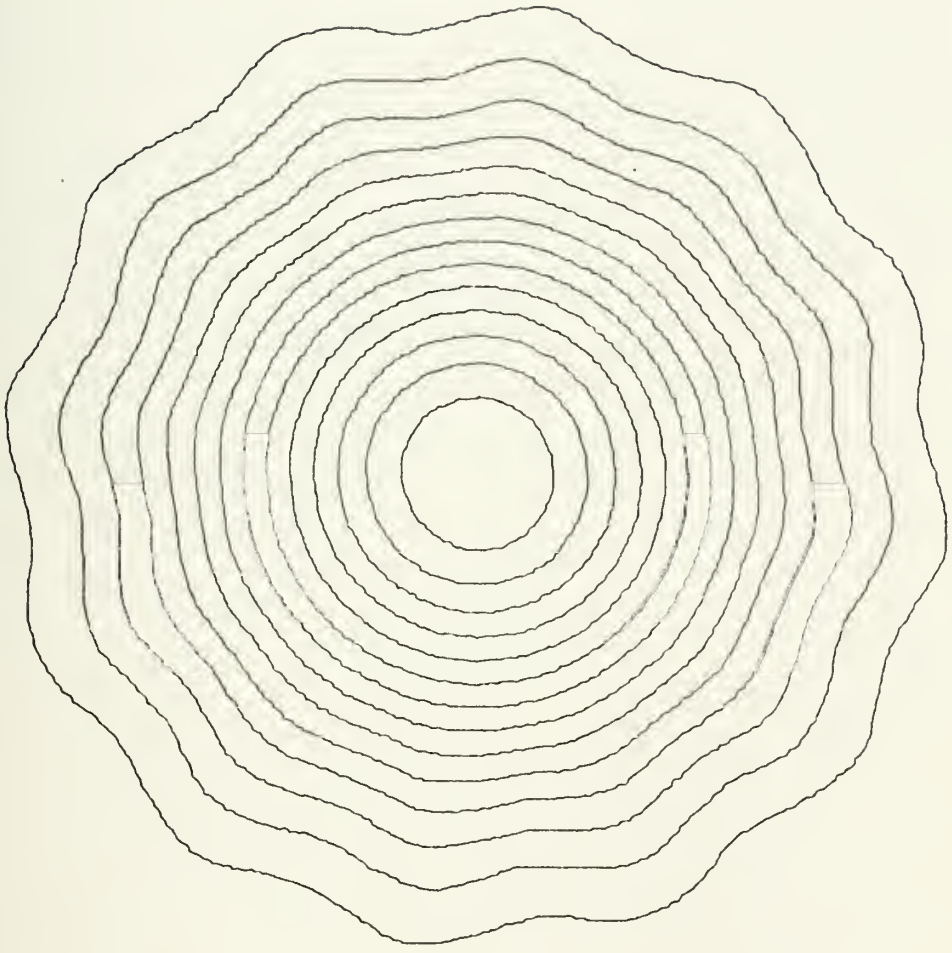

Chart G. Initial PHI field analysis, wave number 12, phase speed 10°/day, $A^* = 2.3 \times 10^7$.

Chart H. 12-hour PHI field forecast, wave number 12, phase speed 10°/day, $A^* = 2.3 \times 10^7$.

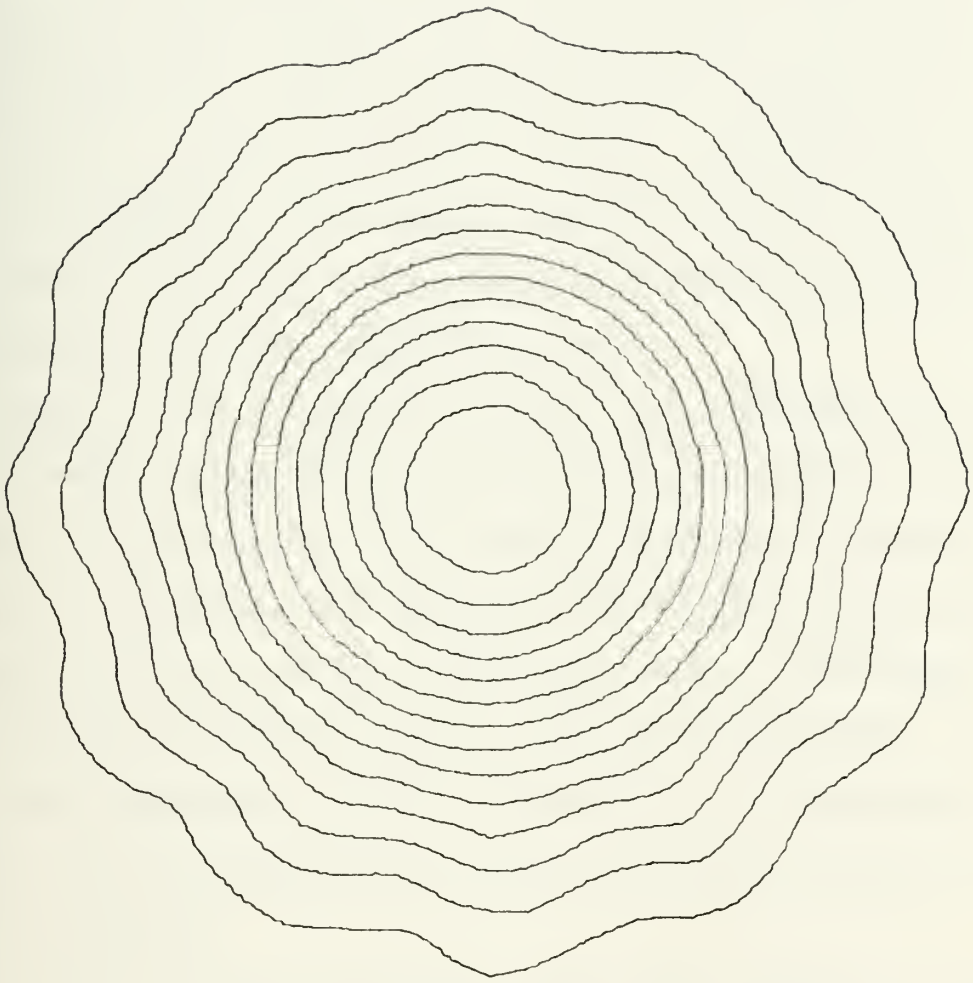

Chart I. 24-hour PHI field forecast, wave number 12, phase speed 10°/day, $A^* = 2.3 \times 10^7$.

IX. CONCLUSIONS

This model showed that the finite element method is applicable to a meteorological system and is competitive with finite difference schemes. The size of the time step that can be taken by a finite element method is comparable with the one for a finite difference scheme. The accuracy of the phase propagation of the finite element method was closer to the analytic solution than second and fourth order finite differencing for the comparisons made.

Based on the increased accuracy and comparable time steps, future research should be continued on the finite element method. Specifically, more advanced methods of solving the system of equations are available and should be utilized, for example, ADI or a cyclic reduction method. Also higher order polynomials should be used for the basis functions. In fact, the real advantage of finite elements is not realized until higher order polynomials are used, from which greater accuracy can be expected.

Research should continue in the polar regions to ascertain the ability of the method to resolve flow over the poles. This will clearly improve the model as it will have realistic physical conditions at the poles instead of artificial boundary conditions. To extend the allowable time step, the equations should be written so that the three equations would be coupled at any one time step. Also the instability

observed in the equatorial regions after long time integrations should be further investigated.

The barotropic model should be expanded into a multi-level baroclinic model with a heating package. Real data should also be applied to the barotropic and baroclinic models to test the method's ability to handle actual conditions.

Finally, research should be performed using smaller elements perhaps selectively in certain areas. With the success with the icosahedral grid, the application of the finite element method to fine meshed models shows promise to provide greater resolution in specific regions.

APPENDIX A

EQUATION FORMULATION

In the initial stages of this paper several decisions were made as to the order and method of solving the three coupled equations. It was decided to uncouple each equation so that it would be one equation with one unknown. Each equation would be written in the time extrapolated Crank-Nicholson method. Although this method has a slower theoretical convergence rate than solving the three equations coupled together, the number of computations per time step is reduced. If the three equations are coupled, then each equation's solution after one iteration at one time step is used in solving the other two equations for the same iteration for the same time step. This requires area integration of each equation for each iteration at any one time step. For example, with the Crank-Nicholson method the ϕ in the u-equation is written at time $N+\frac{1}{2}$. Since this is an unknown, each iteration of the ϕ equation for time, N+1, gives a different $\phi^{N+\frac{1}{2}}$. The pressure gradient term in the u-equation would have to be integrated again as it is not the same as it was at the last iteration. The uncoupling of the three equations was chosen because it sharply reduces the number of computations per time step. The advantage of coupling the three equations at any one time step would be that the equations would be more accurate and consistent and larger

time steps would be possible. Even with the restriction on the size of the time step, a 12-minute time step was possible before instability was experienced.

The first equation solved during one time step is the continuity equation

$$\frac{\partial \phi}{\partial t} + \frac{1}{a \cos \theta} \frac{\partial}{\partial \lambda} (\phi u) + \frac{1}{a \cos \theta} \frac{\partial}{\partial \theta} (\phi v \cos \theta) = 0 \qquad \text{(A-1)}$$

An inner product is now performed on each term with a global pyramid function V_i. The inner product is defined

$$\langle f(\lambda,\theta), V_i \rangle \equiv \iint_{\text{global}} f(\lambda,\theta)\, V_i\, a^2 \cos \theta \, d\lambda d\theta \qquad \text{(A-2)}$$

The a^2 is not carried since it is a common factor and can be cancelled now.

The continuity equation becomes

$$\left\langle \frac{\partial \phi}{\partial t}, V_i \right\rangle + \frac{1}{a} \left\langle \frac{1}{\cos \theta} \frac{\partial}{\partial \lambda} (\phi u), V_i \right\rangle + \frac{1}{a} \left\langle \frac{1}{\cos \theta} \frac{\partial}{\partial \theta} (\phi v \cos \theta), V_i \right\rangle = 0 \qquad \text{(A-3)}$$

If the second and third terms are integrated by parts, the continuity equation becomes

$$\left\langle \frac{\partial \phi}{\partial t}, V_i \right\rangle + \frac{a^2}{a} \int (\phi u V_i) \Big|_{\lambda_0}^{2\pi} d\theta - \frac{1}{a} \left\langle \frac{\phi u}{\cos \theta}, \frac{\partial V_i}{\partial \lambda} \right\rangle + \frac{a^2}{a} \int (\phi v \cos \theta\, V_i) \Big|_{\theta}^{-\frac{\pi}{2}}{}^{\frac{\pi}{2}} d\lambda - \frac{1}{a} \left\langle \frac{\phi v \cos \theta}{\cos \theta}, \frac{\partial V_i}{\partial \theta} \right\rangle = 0 \,. \qquad \text{(A-4)}$$

The second term in the above equation is zero due to cyclic continuity and the fourth term is zero due to the value of cos θ at $-\frac{\pi}{2}$ and $\frac{\pi}{2}$. The continuity equation becomes

$$\left\langle \frac{\partial \phi}{\partial t}, V_i \right\rangle - \frac{1}{a} \left\langle \frac{\phi u}{\cos \theta}, \frac{\partial V_i}{\partial \lambda} \right\rangle - \frac{1}{a} \left\langle \frac{\phi v \cos \theta}{\cos \theta}, \frac{\partial V_i}{\partial \theta} \right\rangle = 0 \qquad \text{(A-5)}$$

To extrapolate the dependent variable forward in time the Crank-Nicholson method first uses a simple forward difference in time to N+1 followed by an average at time level N+1 with values at time level N for the space derivatives. The space derivatives, therefore, are evaluated at time level N+½. Since the three equations were uncoupled, any variable that is not the variable for the particular equation is evaluated at time N+½ while the variable for the equation is evaluated at times N and N+1. The continuity equation becomes

$$\left\langle \left(\frac{\phi^{N+1} - \phi^{N}}{\Delta t} \right), V_i \right\rangle - \frac{1}{2a} \left\langle \frac{\phi^{N+1} u^{N+\frac{1}{2}}}{\cos \theta}, \frac{\partial V_i}{\partial \lambda} \right\rangle - \frac{1}{2a} \left\langle \frac{\phi^{N} u^{N+\frac{1}{2}}}{\cos \theta}, \frac{\partial V_i}{\partial \lambda} \right\rangle$$

$$- \frac{1}{2a} \left\langle \frac{\phi^{N+1} v^{N+\frac{1}{2}}}{\cos \theta} \cos \theta, \frac{\partial V_i}{\partial \theta} \right\rangle - \frac{1}{2a} \left\langle \frac{\phi^{N} v^{N+\frac{1}{2}}}{\cos \theta} \cos \theta, \frac{\partial V_i}{\partial \theta} \right\rangle = 0 \qquad \text{(A-6)}$$

The variables u and v at time N+½ are represented by the second order approximation

$$u^{N+\frac{1}{2}} = u^* \approx \frac{3}{2} u^{N} - \frac{1}{2} u^{N-1} \qquad \text{(A-7)}$$

$$v^{N+\frac{1}{2}} = v^* \approx \frac{3}{2} v^{N} - \frac{1}{2} v^{N-1} \qquad \text{(A-8)}$$

This completes the uncoupling and the continuity equation becomes

$$<(\phi^{N+1}-\phi^{N}),V_i> - \frac{\Delta t}{2a}\left[<\frac{\phi^{N+1}u^*}{\cos\theta},\frac{\partial V_i}{\partial\lambda}> + <\frac{\phi^{N+1}v^*}{\cos\theta}\cos\theta,\frac{\partial V_i}{\partial\theta}>\right]$$

$$- \frac{\Delta t}{2a}\left[<\frac{\phi^{N}u^*}{\cos\theta},\frac{\partial V_i}{\partial\lambda}> + <\frac{\phi^{N}v^*}{\cos\theta}\cos\theta,\frac{\partial V_i}{\partial\theta}>\right] = 0 \qquad \text{(A-9)}$$

The continuity equation is now one equation in one unknown.

Using Galerkin formulation with Einsteinian notation,† the variables u, v and ϕ are represented by

$$u = \alpha_j(t)\ V_j \qquad \text{(A-10)}$$

$$v = \beta_j(t)\ V_j \qquad \text{(A-11)}$$

$$\phi = \gamma_j(t)\ V_j \qquad \text{(A-12)}$$

where V_j are the low order polynomials that are the basis functions. In this paper we used linear polynomials of the form

$$f = d + b\lambda + e\theta \qquad \text{(A-13)}$$

The coefficients α, β and γ are the scalar values of the variables and are functions of time only.

---

†A repeated index implies summation with respect to that index.

Using this new notation, the continuity equation becomes

$$< (\gamma_j^{N+1}-\gamma_j^N) V_j, V_i > - \frac{\Delta t}{2a}\left[<\gamma_j^{N+1} \frac{\alpha_k^* V_j V_k}{\cos\theta}, \frac{\partial V_i}{\partial \lambda} > \right.$$

$$\left. + <\gamma_j^{N+1}\beta_k^* \frac{\cos\theta}{\cos\theta} V_j V_k, \frac{\partial V_i}{\partial \theta}>\right] - \frac{\Delta t}{2a}\left[<\gamma_j^N \frac{\alpha_k^* V_j V_k}{\cos\theta}, \frac{\partial V_i}{\partial \lambda}>\right.$$

$$\left. + <\gamma_j^N \beta_k^* \frac{\cos\theta}{\cos\theta} V_j V_k, \frac{\partial V_i}{\partial \theta}>\right] = 0 \qquad \text{(A-14)}$$

where V_i is the pyramid function at the point where the equation is being written. It is now clear why the pyramid function was introduced. This procedure is repeated for each grid point leading to a system of equations for the value of dependent variable γ_j^{N+1} at the N grid points.

The trigonometric functions in the equation can be handled in several ways. Cullen (1974) treated the trigonometric functions as constants over each triangle. As the grid length decreases the accuracy of this approximation increases. The decision was made to interpolate the trigonometric functions into the Galerkin space. The trigonometric functions were represented by

$$\cos\theta = \xi_j V_j \qquad \text{(A-15)}$$

$$\sin\theta = \eta_j V_j \qquad \text{(A-16)}$$

For ease of notation the trigonometric functions were cancelled where possible from the inner products and we now define a new inner product

$$\underline{\langle u, V_i \rangle} = \int_0^2 \int_{-\pi/2}^{\pi/2} u\, V_i \, d\lambda d\theta \qquad \text{(A-17)}$$

The continuity equation becomes

$$\underline{\langle (\gamma_j^{N+1} - \gamma_j^N)\, \xi_k V_j V_k, V_i \rangle} - \frac{\Delta t}{2a}\left[\underline{\langle \gamma_j^{N+1} \alpha_k^* V_j V_k, \frac{\partial V_i}{\partial \lambda} \rangle}\right.$$

$$\left. + \underline{\langle \gamma_j^{N+1} \beta_k^* \xi_L V_j V_k V_L, \frac{\partial V_i}{\partial \theta} \rangle}\right] - \frac{\Delta t}{2a}\left[\underline{\langle \gamma_j^N \alpha_k^* V_j V_k, \frac{\partial V_i}{\partial \lambda} \rangle}\right.$$

$$\left. + \underline{\langle \gamma_j^N \beta_k^* \xi_L V_j V_k V_L, \frac{\partial V_i}{\partial \theta} \rangle}\right] = 0 \qquad \text{(A-18)}$$

Since everything is known in this equation, with the exception of γ_j^{N+1}, and the γ_j are functions of time only, we can take the γ_j outside the inner products and write the following matrix equation

$$\gamma_j^{N+1} \left[\bar{A} - \frac{\Delta t}{2a} \bar{B}\right] = \gamma_j^N \left[\bar{A} + \frac{\Delta t}{2a} \bar{B}\right] \qquad \text{(A-19)}$$

where

$$\bar{A} = \underline{\langle \xi_k V_j V_k, V_i \rangle}$$

and

$$\bar{B} = \underline{\langle \alpha_k^* V_j V_k, \frac{\partial V_i}{\partial \lambda} \rangle} + \underline{\langle \beta_k^* \xi_L V_j V_k V_L, \frac{\partial V_i}{\partial \theta} \rangle} \qquad \text{(A-19.2)}$$

The matrix equation is manipulated to solve for the change, e_j^N , in γ_j

$$\gamma_j^{N+1}[\bar{A} - \frac{\Delta t}{2a}\bar{B}] = \gamma_j^N[\bar{A} + \frac{\Delta t}{2a}\bar{B}] + \gamma_j^N(\frac{\Delta t}{2a}\bar{B} - \frac{\Delta t}{2a}\bar{B}) \quad \text{(A-20.1)}$$

$$\gamma_j^{N+1}[\bar{A} - \frac{\Delta t}{2a}\bar{B}] = \gamma_j^N[\bar{A} - \frac{\Delta t}{2a}\bar{B}] + \gamma_j^N \frac{\Delta t}{a}\bar{B} \quad \text{(A-20.2)}$$

$$e_j^N[\bar{A} - \frac{\Delta t}{2a}\bar{B}] = \gamma_j^N \frac{\Delta t}{a}\bar{B} \quad \text{(A-20)}$$

where

$$\gamma_j^{N+1} = \gamma_j^N + e_j^N \quad \text{(A-20.3)}$$

The zonal equation was developed in the same manner as the continuity equation and the matrix equation derived was

$$e_j^N[\bar{A} + \frac{\Delta t}{2a}\bar{D}] = \frac{\Delta t}{a}\bar{E} - \alpha_j^N \frac{\Delta t}{a}\bar{D} \quad \text{(A-21)}$$

where

$$\bar{D} = \underline{<}\alpha_k^* V_k \frac{\partial V_j}{\partial \lambda}, V_i\underline{>} + \underline{<}\beta_k^* \xi_L V_k V_L \frac{\partial V_j}{\partial \theta}, V_i\underline{>} \quad \text{(A-21.1)}$$

and

$$\bar{E} = \underline{<}2a\Omega\beta_k^* \xi_L \eta_M V_k V_L V_M, V_i\underline{>} - \underline{<}\gamma_k^* \frac{\partial V_k}{\partial \lambda}, V_i\underline{>}$$

$$+ <\alpha_k^* \beta_j^* \eta_L V_k V_j V_L, V_i\underline{>} \quad \text{(A-21.2)}$$

The meridional equation becomes

$$e_j^N[\bar{A} + \frac{\Delta t}{2a}\bar{F}] = -\beta_j^N \frac{\Delta t}{a}\bar{F} - \frac{\Delta t}{a}\bar{G} \qquad \text{(A-22)}$$

where

$$\bar{F} = \langle \alpha_k^* V_k \frac{\partial V_j}{\partial \lambda}, V_i \rangle + \langle \beta_k^* \xi_L V_k V_L \frac{\partial V_j}{\partial \theta}, V_i \rangle \qquad \text{(A-22.1)}$$

and

$$\bar{G} = \langle \alpha_k^* \alpha_L^* \eta_M V_k V_L V_M, V_i \rangle + \langle 2a\Omega \alpha_k^* \xi_L \eta_M V_k V_L V_M, V_i \rangle$$

$$+ \langle \gamma_k^* \xi_L \frac{\partial V_k}{\partial \theta} V_L, V_i \rangle \qquad \text{(A-22.2)}$$

The matrices in equations (A-20), (A-21), and (A-22) are built by the procedures in Section III.C.1. Once the matrices for each equation have been built, the system of equations derived can be solved by any conventional process. A simple Gauss-Seidel iterative procedure was chosen to solve the matrix equations with a relative improvement check used as a cutoff criterion. Subroutine SOLVE in the Computer Program section solved the equations.

While the purpose of this paper was to develop a finite element barotropic primitive equation model, it was hoped that a considerable savings in time could be realized. Further time savings can be achieved if more sophisticated techniques such as, Successive Over Relaxation, SOR, or Alternating

Direction Implicit, ADI, are used. Leslie and McAvaney (1973) show that the computing time can be greatly reduced by using some of the newer direct techniques. These latter were not employed during development since programs for these methods were not available in the library of programs.

APPENDIX B

ICOSAHEDRAL GRID FORMULATION

A spherical icosahedron is made up of twenty equilateral spherical triangles. Each interior angle is $2\pi/5$. The method of subdividing the icosahedron is accomplished in three steps. First, the top five major spherical triangles are partitioned. Second, the Southern Hemisphere triangles are reflected from the Northern Hemisphere. The last step is the subdivision of the equatorial triangles.

The first Northern Hemisphere major spherical triangle is subdivided and the next four adjacent major spherical triangles are given the same solution for longitude and latitude of the nodes except that the longitudes are displaced the proper multiple of $2\pi/5$ radians. In all cases, the purpose of the subdivisions is twofold. The main purpose is to specify the latitude and longitude of every node. The second purpose is to assign a global correspondence number to each node.

The second step in subdividing the sphere is to mirror the Northern Hemisphere spherical triangles into the Southern Hemisphere. The solutions are the same except that the longitudes are displaced $\pi/5$ radians.

The third step is the solution of the ten major spherical, equatorial triangles. The following description shows how one Northern Hemisphere triangle, two equatorial triangles and one Southern Hemisphere triangle can equally be subdivided

and solutions for longitude and latitude of each node be obtained (Figure 14). Three laws of spherical trigonometry are required. They are:

Law of Sines

$$\frac{\sin a}{\sin A} = \frac{\sin b}{\sin B} \qquad \text{(B-1)}$$

Law of Cosines for Sides

$$\cos a = \cos b \cos c + \sin b \sin c \cos A \qquad \text{(B-2)}$$

Law of Cosines for Angles

$$\cos A = -\cos B \cos C + \sin B \sin C \cos a \qquad \text{(B-3)}$$

where a represents a side (spherical arc) of a triangle and A represents the corresponding opposite angle.

Since all three angles of a triangle are known, the length of arc a can be determined by the law of cosines for sides. Since a is along a meridian, the length of this arc is the colatitude of point 2. The longitude of points 1 and 2 is arbitrarily chosen as zero radians. Arc a is then divided into n segments. The latitudes and longitudes of these points can easily be computed. Arc b can be segmented in the same manner. Arc d can be found by using the law of sines with angle C and arcs 3-1 and 4-1. Arc d is then segmented into

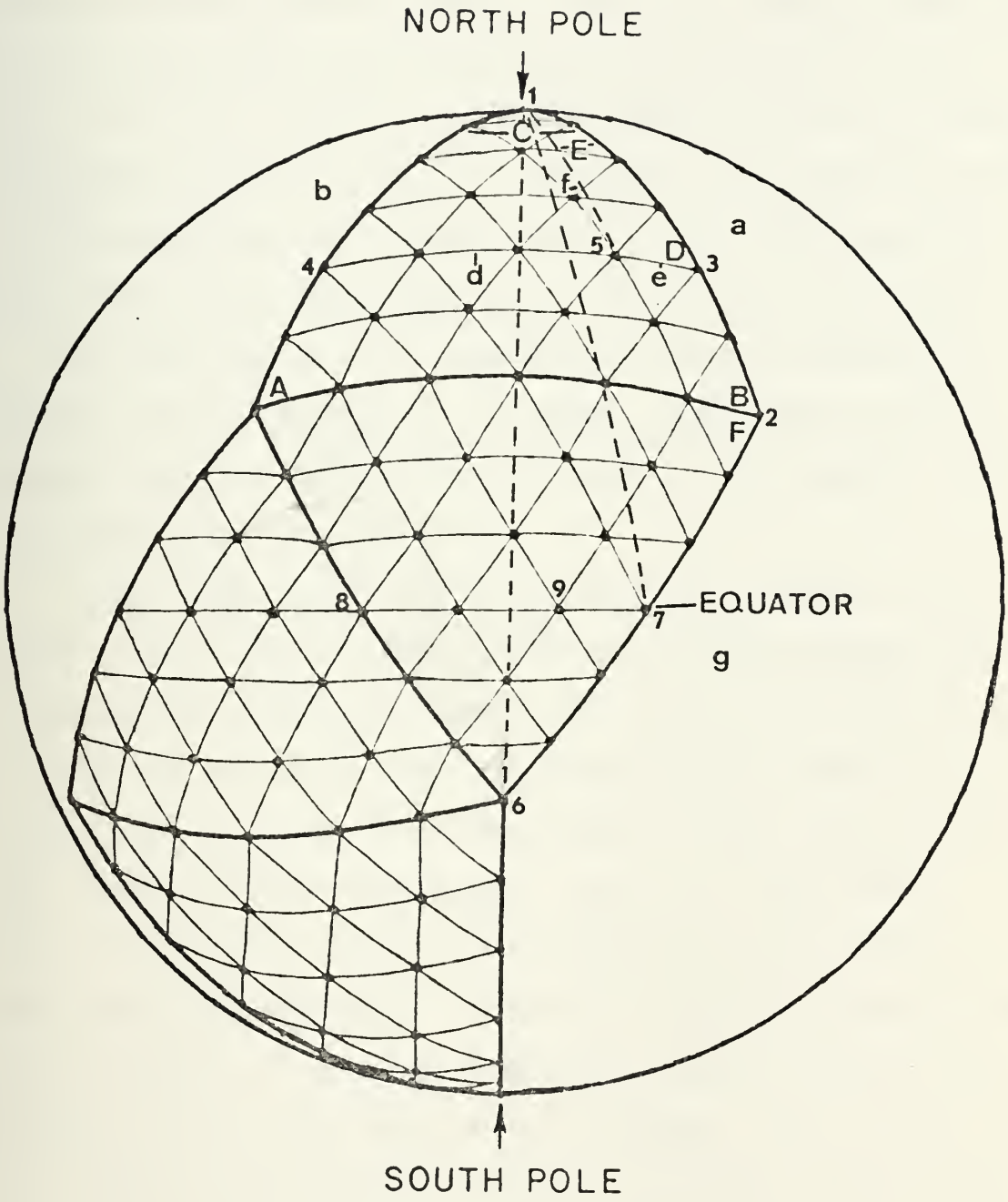

FIGURE 14. One Northern Hemispheric, two Equatorial and one Southern Hemispheric major spherical triangle.

the proper number of pieces. The three sides of triangle 1-4-3 are known, therefore angle D can be found by the law of cosines for sides. Arc 1-3, arc e and angle D with the law of sines determine the colatitude of point 5, arc f. Arc 1-3, arc e and arc f with the law of cosines for sides determine angle E, point 5's longitude. All points in the Northern Hemisphere triangle can be solved for longitude and latitude by manipulating these laws, arcs and angles.

All the points in a Southern Hemisphere triangle are mirror images of those in a Northern Hemisphere triangle except that the longitudes are displaced $\pi/5$ radians and the latitudes have a minus sign.

With angle (B+F), arc 1-6, angle C/2, and the law of sines, arc g can be found. This arc is then divided into n segments. With angle (B+F), arc a, arc 2-7, and the law of cosines for sides, arc 1-7 can be found. This arc is the colatitude of point 7 but care must be taken as this arc can extend across the equator. Arc 1-7, angle (B+F), arc (2-7) and the law of sines determine angle (2-1-7), the longitude of point 7. Point 8's longitude and latitude can be found by a similar computation using the arcs and angles on the other side of the triangle. Arc 7-8 can now be found and subdivided. Angle (8-7-1) can also be found. Then triangle (9-7-1) can be solved to determine the latitude and longitude of point 9. The rest of the points in both major spherical equatorial triangles can be solved for latitude and longitude in a similar manner. The Computer Program Section

contains a program written in Fortran IV for an IBM 360 computer which will subdivide an icosahedron simply by giving it the desired number of segments, n.

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C       THIS IS THE MAIN BODY OF THE FINITE ELEMENT    C
C       PROGRAM.                                       C
C       THE INTEGER*2 STATEMENTS ARE USED TO STORE     C
C       THE REQUIRED TABLES USED DURING INTEGRATION.   C
C       SINCE THE LARGEST INTEGER STORED DID NOT       C
C       EXCEED THE CAPABILITIES OF THE INTEGER*2 MODE  C
C       IT WAS USED TO CONSERVE CORE REQUIREMENTS.     C
C       MATRICES A, B, C AND D CONTAIN THE RESULTS     C
C       OF THE GLOBAL INTEGRATION PERFORMED ON EACH    C
C       TERM IN THE EQUATION. MATRICES A AND B ARE USED C
C       DURING THE GAUSS-SIEDAL ITERATION ON THE U AND C
C       V EQUATIONS. MATRICES C AND D ARE USED DURING  C
C       THE ITERATION ON THE GEOPOTENTIAL FIELD.       C
C       VECTORS                                        C
C            AREA2= TWICE THE AREA OF EACH TRIANGLE    C
C            A1 =A CONSTANT OVER EACH TRIANGLE USED TO C
C                 EVALUATE A LATITUDE DERIVATIVE.      C
C            A2= A CONSTANT SIMILIAR IN NATURE TO A1   C
C            A3= SAME AS A1                            C
C            B1= A CONSTANT USED TO EVALUATE A LONGITUDE C
C                 DERIVATIVE.                          C
C            B2= SAME AS B1                            C
C            B3= SAME AS B1                            C
C            XLAT = LATITUDES OF ALL THE POINTS        C
C            XLON= LONGITUDES OF ALL THE POINTS        C
C            XLON1= LONGITUDES OF THE POINTS THAT ARE  C
C                 USED TO HAVE CYCLIC CONTINUITY       C
C            COSLAT= COSINE OF ALL THE LATITUDES OF EACH C
C                 POINT                                C
C            SINLAT= SINE OF ALL THE LATITUDES OF EACH C
C                 POINT                                C
C            NTRI= NUMBER OF THE TRIANGLE WHERE CYCLIC C
C                 CONTINUITY IS MAINTAINED THRU USE OF C
C                 VECTOR XLON1 VICE XLON. ON ONE SIDE  C
C                 OF THE CYCLIC CONTINUITY LINE THE    C
C                 TRIANGLES HAVE 360 DEGREES AS A LONG- C
C                 ITUDE WHILE FROM THE OTHER SIDE THE  C
C                 TRIANGLE EXPERIENCES 0 DEGREES.      C
C            NTRO= NUMBER OF THE POINT WHERE CYCLIC    C
C                 CONTINUITY IS MAINTAINED             C
C            TERM2= TEMPORARY HOLDING VECTOR USED DURING C
C                 ITERATION TO MINIMIZE THE COMPUTATION C
C                 OF THE RIGHT HAND SIDE OF THE EQUATION C
C            SPH= A DUMMY VECTOR USED WHEN A TRIG FUNC- C
C                 TION IS NOT BEING INTEGRATED INTO THE C
C                 GALERKIN SPACE BY A SUBROUTINE       C
C            EN= THE CHANGE IN THE VARIABLE DURING ONE C
C                 TIME STEP. THIS IS THE VECTOR THAT THE C
C                 ITERATION SCHEME CONVERGES TO.       C
C            NTRI5= THE NUMBER OF THE TRIANGLES THAT ARE C
C                 SUPPORTED BY ONLY FIVE TRIANGLES VICE C
C                 SIX TRIANGLES. THIS OCCURS AT THE    C
C                 VERTEXES OF THE MAJOR SPHERICAL TRI- C
C                 ANGLES.                              C
C            VECT= TEMPORARY HOLDING VECTOR USED DURING C
C                 THE PHI EQUATION.                    C
C            U1,V1,PHI1= INITIAL FIELDS AND FIELDS AT  C
C                 THE TIME LEVEL N                     C
C            U2,V2,PHI2= FIELDS AT TIME LEVEL N+1      C
C            USTAR=TIME EXTRAPOLATED ASSUMPTION ON U   C
C            VSTAR=TIME EXTRAPOLATED ASSUMPTION ON V   C
C            PHSTAR=TIME EXTRAPOLATED ASSUMPTION ON PHI C
```

```
C              E,F AND G= HOLDING VECTORS FOR THE TANGENT,  C
C                   CORIOLIS AND PRESSURE GRADIENT TERMS    C
C              PHI3= HOLDING VECTOR USED FOR HARMONIC       C
C                   ANALYSIS                                C
C              COA=COEFFICIENT OF A FOR EACH TRIANGLE IN    C
C                   EXPRESSION F=A+B*LAMBDA+C*THETA         C
C              COB=COEFFICIENT FOR B FOR EACH TRIANGLE      C
C              BFLD=GLOBAL 73 X 35 5 DEGREE GRID            C
C              FLD=NORTHERN HEMISPHERIC GRID USED FOR PLOT  C
C              CL=CONTOUR LEVELS FOR PLOT                   C
C         TABLES                                            C
C              NPTS=GLOBAL CORRESPONDENCE TABLE             C
C              NCOR=GLOBAL CORRELATION TABLE                C
C              NTABL=TABLE TO CONVERT FROM ICOSAHEDRAL      C
C                   GRID TO 73 X 35 GRID.                   C
C              COC=COEFFICIENT FOR C FOR EACH TRIANGLE      C
 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      INTEGER*2 NPTS,NCOR,NTRI5
      INTEGER*2 NTABL
      COMMON /CM1/ AREA2(1296),A1(1296),A2(1296),A3(1296),
     1B1(1296),B2(1296),B3(1296)
      COMMON /CM2/ XLAT(684),XLON(684),XLON1(684)
      COMMON /CM3/ NPTS(1300,3)
      COMMON /CM4/ COSLAT(684),SINLAT(684)
      COMMON /CM5/ NCOR(684,7)
      COMMON /CM6/ NTRI(46),NTRO(23)
      COMMON /CM7/ TERM2(684)
      DIMENSION A(684,7),B(684,7),C(684,7),D(684,7)
      DIMENSION SPH(684),EN(684),NTRI5(12),VECT(684)
      DIMENSION U1(684),U2(684),V1(684),V2(684)
      DIMENSION PHI1(684),PHI2(684)
      DIMENSION USTAR(684),VSTAR(684),PHSTAR(684)
      DIMENSION E(684),F(684),G(684)
      DIMENSION PHI3(36)
      DIMENSION NTABL(72,18)
      DIMENSION COA(640),COB(640),COC(640)
      DIMENSION BFLD(73,35),FLD(63,63)
      DIMENSION CL(16)
      DATA DT2/360./,DT/ 720./,N/ 8/,OMEGA/7.292115432E-05/
      DATA WAVENO/ 8./,WAVVEL/1.61604E-05/,AMP/ 5.49364307/
      DATA IFLAG/0/,TIME/0.0/
      LOGICAL*1 LTG(3)/3*.FALSE./
      REAL*8 TITLE1(12)/' HINSMAN',' D.E.  ','INITIAL '
     1,'CONDITIO','NS WAVEN','UMBER 8 ','PHASE SP',
     2'EED 10 D','EGREES/D','AY A=7.E','+07     ',
     3'          '/
      REAL*8 TITLE2(12)/' HINSMAN',' D.E.  ','FORCAST '
     1,'CONDITIO','NS WAVEN','UMBER 8 ','PHASE SP',
     2'EED 10 D','EGREES/D','AY A=7.E','+07     ',
     3'          '/
      REAL*8 SUBTIT(12)/'      12',','      24',','      36',
     1'      48',','      60',','      72',
     2'      84',','      96',','     108',','     120',
     3'     132',','     144'/
      NOTRI=1280
      NOPTS=642
      J1=0
      JM=18
      Z=8.
      READ(5,100) XLAT
      READ(5,100) XLON
      DO 3 I=1,NOTRI,5
      L=I+4
    3 READ(5,101)((NPTS(K,1),NPTS(K,3),NPTS(K,2)),K=I,L)
      DO 4 I=1,NOPTS
    4 READ(5,102)(NCOR(I,J),J=1,7)
  100 FORMAT(8F10.6)
  101 FORMAT(5(3I5))
  102 FORMAT(7I10)
  103 FORMAT(10F8.1)
      READ(5,104) NTRI5
  104 FORMAT(10I8)
```

```
      READ(5,106) NTRI
      READ(5,106) NTRO
  106 FORMAT(8I10)
      DO 1 I=1,NOPTS
      DO 1 J=1,7
      A(I,J)=0.0
      B(I,J)=0.0
      C(I,J)=0.0
    1 D(I,J)=0.0
      DO 2 I=1,NOPTS
    2 SPH(I)=1.
      CALL AREAZ(NOTRI,OMEGA,NOPTS)
      CALL SURVEY(NOTRI,NTABL,NOPTS)
      DO 325 I=1,72
      IL=73-I
  325 WRITE(6,326)(NTABL(IL,J),J=1,18)
  326 FORMAT(/1X,18I7)
      CALL SOLUT(XLAT,XLON,U1,V1,PHI1,NOPTS,WAVENO,WAVVEL,
     1AMP,0,0,0,DT,TIME)
      U1(1)=0.0
      U1(NOPTS)=0.0
      V1(1)=0.0
      V1(NOPTS)=0.0
      WRITE(6,130)
  130 FORMAT(/1X,' INITIAL PHI FIELD')
      WRITE(6,105)(PHI1(I),I=1,NOPTS)
      WRITE(6,131)
  131 FORMAT(/1X,' INITIAL U FIELD')
      WRITE(6,105)(U1(I),I=1,NOPTS)
      WRITE(6,132)
  132 FORMAT(/1X,' INITIAL V FIELD')
      WRITE(6,105)(V1(I),I=1,NOPTS)
      DO 50 I=1,16
   50 CL(I)=1.
      CALL EVAL(NOTRI,PHI1,COA,COB,COC)
      CALL DISPL(BFLD,NTABL,COA,COB,COC)
      DO 36 J=1,18
      L=1
      JJ=36-J
      DO 37 I=2,72,2
      PHI3(L)=BFLD(I,JJ)
   37 L=L+1
   36 CALL HARMAN(J1,JM,Z,PHI3,JJ)
      CALL LONGOU(BFLD,FLD,PHI1(1))
      CALL CONTUR(FLD,63,63,63,CL,-16,TITLE1,6,6,LTG)
      DO 5 I=1,NOPTS
      USTAR(I)=U1(I)
      VSTAR(I)=V1(I)
      PHSTAR(I)=PHI1(I)
      VECT(I)=0.0
    5 EN(I)=0.0
  324 FORMAT(////)
      KMAX=60
      DO 71 KL=1,6
      DO 70 KK=1,60
      DO 7 I=1,NOPTS
      E(I)=0.0
      F(I)=0.0
    7 G(I)=0.0
      CALL AMATRI(NOTRI,A,COSLAT)
      DO 700 I=1,NOPTS
      DO 700 J=1,7
  700 C(I,J)=A(I,J)
  501 CALL BMATRI(NOTRI,USTAR,B1,B2,B3,SPH,A,B)
      CALL BMATRI(NOTRI,VSTAR,A1,A2,A3,COSLAT,A,B)
 2000 CALL DMATRI(NOTRI,USTAR,B1,B2,B3,SPH,D)
      CALL DMATRI(NOTRI,VSTAR,A1,A2,A3,COSLAT,D)
      DO 6 I=1,NOPTS
      DO 6 J=1,7
      C(I,J)=C(I,J)-DT2*D(I,J)
    6 D(I,J)=D(I,J)*DT
  202 CALL SOLVE(C,D,EN,PHI1,NOPTS,NTRI5,VECT,IFLAG)
```

```
      DO 8 I=1,NOPTS
    8 PHI2(I)=PHI1(I)+EN(I)
  210 SUM=0.0
      DO 60 I=2,6
   60 SUM=SUM+PHI2(I)
      AVPHI=SUM/5.
      DO 61 I=1,6
   61 PHI2(I)=AVPHI
      SUM=0.0
      DO 62 I=637,641
   62 SUM=SUM+PHI2(I)
      AVPHI=SUM/5.
      DO 63 I=637,642
   63 PHI2(I)=AVPHI
      CALL TANVEC(NOTRI,USTAR,SINLAT,VSTAR,G)
  204 CALL CORIO(NOTRI,VSTAR,E,SINLAT,COSLAT)
      CALL PGF(NOTRI,PHSTAR,B1,B2,B3,SPH,F)
      DO 11 I=1,NOPTS
      E(I)=DT*(E(I)-F(I)+G(I))
   11 EN(I)=0.0
      CALL SOLVE(A,B,EN,U1,NOPTS,NTRI5,E,IFLAG)
      DO 12 I=1,NOPTS
   12 U2(I)=U1(I)+EN(I)
  201 DO 13 I=1,NOPTS
      EN(I)=0.0
      E(I)=0.0
      G(I)=0.0
   13 F(I)=0.0
  207 CALL TANVEC(NOTRI,USTAR,SINLAT,USTAR,G)
      CALL CORIO(NOTRI,USTAR,E,SINLAT,COSLAT)
      CALL PGF(NOTRI,PHSTAR,A1,A2,A3,COSLAT,F)
      DO 15 I=1,NOPTS
   15 G(I)=-DT*(E(I)+F(I)+G(I))
 2001 CALL SOLVE(A,B,EN,V1,NOPTS,NTRI5,G,IFLAG)
      DO 18 I=1,NOPTS
   18 V2(I)=V1(I)+EN(I)
  200 DO 17 I=1,NOPTS
      DO 17 J=1,7
      A(I,J)=0.0
      B(I,J)=0.0
      C(I,J)=0.0
   17 D(I,J)=0.0
      DO 16 I=1,NOPTS
      PHSTAR(I)=1.5*PHI2(I)-.5*PHI1(I)
      USTAR(I)=1.5*U2(I)-.5*U1(I)
      VSTAR(I)=1.5*V2(I)-.5*V1(I)
      PHI1(I)=PHI2(I)
      U1(I)=U2(I)
      V1(I)=V2(I)
   16 EN(I)=0.0
  105 FORMAT(/1X,6E20.6)
   70 CONTINUE
      TIME=FLOAT((KL-1)*KMAX+KK-1)*DT
      TIME1=TIME/3600.
      WRITE(6,310) TIME1
  310 FORMAT(/1X,'PROG      SOLUTION AT',2X,F5.1,2X,'HOURS')
      WRITE(6,122)
  122 FORMAT(/1X,' PHI FIELD')
      WRITE(6,105)(PHI2(I),I=1,NOPTS)
      WRITE(6,110)
  110 FORMAT(/1X,' U FIELD')
      WRITE(6,105)(U2(I),I=1,NOPTS)
      WRITE(6,111)
  111 FORMAT(/1X,' V FIELD')
      WRITE(6,105)(V2(I),I=1,NOPTS)
      CALL EVAL(NOTRI,PHI1,COA,COB,COC)
      CALL DISPL(BFLD,NTABL,COA,COB,COC)
      DO 38 J=1,18
      L=1
      JJ=36-J
      DO 39 I=2,72,2
      PHI3(L)=BFLD(I,JJ)
```

```
 39 L=L+1
 38 CALL HARMAN(J1,JM,Z,PHI3,JJ)
    TITLE2(12)=SUBTIT(KL)
    CALL LONGOU(BFLD,FLD,PHI1(1))
    CALL CONTUR(FLD,63,63,63,CL,-16,TITLE2,6,6,LTG)
 71 CONTINUE
141 FORMAT(/1X,' B MATRIX')
140 FORMAT(/1X,' A MATRIX')
107 FORMAT(/1X,7E15.6)
    STOP
    END



    SUBROUTINE SURVEY(NOTRI,NTABL,NOPTS)
    INTEGER*2 NPTS,NTABL
    COMMON /CM2/ XLAT(684),XLON(684),XLON1(684)
    COMMON /CM3/ NPTS(1300,3)
    COMMON /CM4/ COSLAT(684),SINLAT(684)
    COMMON /CM6/ NTRI(46),NTRO(23)
    DIMENSION S(3),ISAVE(3),NTABL(72,1)
    DATA RAD/6.283185308/,THETA/1.570796327/
    DRAD=RAD/72.
    NOTR=NOTRI/2
    DTHET=THETA-DRAD
    NOPT=341
    DO 1 I=1,72
    XLO=FLOAT(I-1)*DRAD
    YLA=DTHET-DRAD
    DO 2 J=2,18
    IFLAG=0
    IF(J.EQ.18) YLA=YLA+.01
    ISAVE(1)=1
    ISAVE(2)=1
 40 IKOUNT=0
    L=2
    DO 3 K=6,NOTR
    IF(IFLAG.EQ.1.AND.K.EQ.ISAVE(2)) GO TO 3
    ICOUNT=0
    XLOMX=0.0
    XLOMN=2.*RAD
    YLAMX=0.0
    YLAMN=2.*RAD
    LL=0
    IF(NTRI(L).NE.K) GO TO 4
    L=L+1
    LL=1
    DO 5 KL=1,3
    XLOMX=AMAX1(XLOMX,XLON1(NPTS(K,KL)))
  5 XLOMN=AMIN1(XLOMN,XLON1(NPTS(K,KL)))
    GO TO 8
  4 DO 6 KL=1,3
    XLOMX=AMAX1(XLOMX,XLON(NPTS(K,KL)))
  6 XLOMN=AMIN1(XLOMN,XLON(NPTS(K,KL)))
  8 DO 7 KL=1,3
    YLAMX=AMAX1(YLAMX,XLAT(NPTS(K,KL)))
  7 YLAMN=AMIN1(YLAMN,XLAT(NPTS(K,KL)))
    IF(XLO.LT.XLOMX.AND.XLO.GE.XLOMN) ICOUNT=ICOUNT+1
    IF(YLA.LT.YLAMX.AND.YLA.GE.YLAMN) ICOUNT=ICOUNT+1
    IF(ICOUNT.LT.2) GO TO 3
    IKOUNT=IKOUNT+1
    IF(IKOUNT.EQ.2) GO TO 9
    ISAVE(1)=K
    GO TO 3
  9 ISAVE(2)=K
    GO TO 10
  3 CONTINUE
 10 ICOUNT=0
    DO 11 KL=1,3
    DO 12 KK=1,3
```

```
      IF(NPTS(ISAVE(1),KL).NE.NPTS(ISAVE(2),KK)) GO TO 12
      IF(ICOUNT.GT.0) GO TO 13
      IS1=NPTS(ISAVE(1),KL)
      ICOUNT=ICOUNT+1
      GO TO 11
   13 IS2=NPTS(ISAVE(1),KL)
      GO TO 14
   12 CONTINUE
   11 CONTINUE
   14 IF(LL.EQ.0) GO TO 15
      IFLAG=1
      IF(XLON1(IS2)-XLON1(IS1).EQ.0.0) GO TO 40
      SLOPE=(XLAT(IS2)-XLAT(IS1))/(XLON1(IS2)-XLON1(IS1))
      GO TO 16
   15 IFLAG=1
      IF(XLON(IS2)-XLON(IS1).EQ.0.0) GO TO 40
      SLOPE=(XLAT(IS2)-XLAT(IS1))/(XLON(IS2)-XLON(IS1))
   16 IF(LL.EQ.0) GO TO 17
      B=XLAT(IS1)-SLOPE*XLON1(IS1)
      GO TO 18
   17 B=XLAT(IS1)-SLOPE*XLON(IS1)
   18 YLAC=SLOPE*XLO+B
      DO 20 KL=1,3
      IF(IS1.EQ.NPTS(ISAVE(1),KL).OR.IS2.EQ.NPTS(ISAVE(1),
     1KL)) GO TO 20
      INOT1=NPTS(ISAVE(1),KL)
      GO TO 22
   20 CONTINUE
   22 DO 23 KL=1,3
      IF(IS1.EQ.NPTS(ISAVE(2),KL).OR.IS2.EQ.NPTS(ISAVE(2),
     1KL)) GO TO 23
      INOT2=NPTS(ISAVE(2),KL)
      GO TO 24
   23 CONTINUE
   24 IF(INOT1.GT.INOT2) GO TO 25
      MP=ISAVE(1)
      MM=ISAVE(2)
      GO TO 26
   25 MP=ISAVE(2)
      MM=ISAVE(1)
   26 IF(YLA.GT.YLAC) NTABL(I,J)=MP
      IF(YLA.LE.YLAC) NTABL(I,J)=MM
    2 YLA=YLA-DRAD
    1 CONTINUE
      DO 30 I=1,15
   30 NTABL(I,1)=5
      DO 31 I=16,29
   31 NTABL(I,1)=4
      DO 32 I=30,44
   32 NTABL(I,1)=3
      DO 33 I=45,58
   33 NTABL(I,1)=2
      DO 34 I=59,72
   34 NTABL(I,1)=1
      NTABL(72,16)=559
      NTABL(71,17)=639
      NTABL(72,17)=639
      NTABL(71,18)=640
      NTABL(72,18)=639
      RETURN
      END



      SUBROUTINE SOLVE(A,B,EN,Z,NOPTS,NTRI5,C,IFLAG)
      INTEGER*2 NCOR,NTRI5
      COMMON /CM5/ NCOR(684,7)
      COMMON /CM7/ TERM2(684)
      COMMON /CM8/ VIS(684)
      DIMENSION A(684,1),B(684,1),EN(1),Z(1),NTRI5(1)
```

```
      DIMENSION C(1)
      DATA EPSI/.1E-05/
      KL=2
      NOPT=NOPTS-1
   22 LX=2
      DO 10 I=KL,NOPT
      LY=7
      IF(I.NE.NTRI5(LX)) GO TO 11
      LY=6
      LX=LX+1
   11 TERM2(I)=0.0
      DO 12 J=1,LY
   12 TERM2(I)=TERM2(I)+Z(NCOR(I,J))*B(I,J)
   10 CONTINUE
      DO 1 L=1,100
      ZMAX=0.0
      ERROR=0.0
      LLL=2
      DO 2 I=KL,NOPT
      LL=7
      IF(I.NE.NTRI5(LLL)) GO TO 4
      LL=6
      LLL=LLL+1
    4 RHS=0.0
      DO 3 J=1,LL
      IF(NCOR(I,J).EQ.I) GO TO 5
      TERM1=EN(NCOR(I,J))*A(I,J)
      RHS=RHS-TERM1
      GO TO 3
    5 K=J
    3 CONTINUE
      RHS=RHS+TERM2(I)+C(I)
      ZDUM=EN(NCOR(I,K))
      ZMAX=AMAX1(ZMAX,ABS(EN(NCOR(I,K))))
   20 EN(NCOR(I,K))=RHS/A(I,K)
      IF(IFLAG.NE.2) GO TO 21
      EN(NCOR(I,K))=EN(NCOR(I,K))*VIS(I)
   21 DELTA=ABS(ZDUM-EN(NCOR(I,K)))
    2 ERROR=AMAX1(ERROR,DELTA)
      EPS=ZMAX*EPSI
      IF(ERROR.LE.EPS) GO TO 6
    1 CONTINUE
    6 WRITE(6,100) L
  100 FORMAT(/1X,I4,2X,'ITERATIONS')
      IF(L.EQ.100) IFLAG=3
      RETURN
      END



      SUBROUTINE BMATRI(NOTRI,DUM,C1,C2,C3,SPH,A,B)
      INTEGER*2 NPTS
      COMMON /CM3/ NPTS(1300,3)
      DIMENSION DUM(1),C1(1),C2(1),C3(1),SPH(1),A(684,1),
     1B(684,1)
      DATA EARTH/6.371E+06/,DT2/360./,DT/ 720./
      DO 1 I=1,NOTRI
      II=NPTS(I,1)
      JJ=NPTS(I,2)
      KK=NPTS(I,3)
      CALL SEARCH(II,JJ,KK,I1,I2,I3,II)
      CALL SEARCH(II,JJ,KK,J1,J2,J3,JJ)
      CALL SEARCH(II,JJ,KK,K1,K2,K3,KK)
      FACT=1./(EARTH*120.)
      TERMA=6.*DUM(II)*SPH(II)+2.*DUM(II)*SPH(JJ)
      TERMB=2.*DUM(II)*SPH(KK)+2.*DUM(JJ)*SPH(II)
      TERMC=2.*DUM(JJ)*SPH(JJ)+DUM(JJ)*SPH(KK)
      TERMD=2.*DUM(KK)*SPH(II)+DUM(KK)*SPH(JJ)+2.*DUM(KK)*SP
     1H(KK)
      TERM1=(TERMA+TERMB+TERMC+TERMD)*FACT
```

```
      TERMB=2.*DUM(II)*SPH(II)+2.*DUM(II)*SPH(JJ)+DUM(II)*SP
     1H(KK)
      TERMC=2.*DUM(JJ)*SPH(II)+6.*DUM(JJ)*SPH(JJ)+2.*DUM(JJ)
     1*SPH(KK)
      TERMD=DUM(KK)*SPH(II)+2.*DUM(KK)*SPH(JJ)+2.*DUM(KK)*SP
     1H(KK)
      TERM2=(TERMB+TERMC+TERMD)*FACT
      TERMC=2.*DUM(II)*SPH(II)+DUM(II)*SPH(JJ)+2.*DUM(II)*SP
     1H(KK)
      TERMD=DUM(JJ)*SPH(II)+2.*DUM(JJ)*SPH(JJ)+2.*DUM(JJ)*SP
     1H(KK)
      TERME=2.*DUM(KK)*SPH(II)+2.*DUM(KK)*SPH(JJ)+6.*DUM(KK)
     1*SPH(KK)
      TERM3=(TERMC+TERMD+TERME)*FACT
      A(II,I1)=A(II,I1)+DT2*TERM1*C1(I)
      B(II,I1)=B(II,I1)-DT*TERM1*C1(I)
      A(II,I2)=A(II,I2)+DT2*TERM1*C2(I)
      B(II,I2)=B(II,I2)-DT*TERM1*C2(I)
      A(II,I3)=A(II,I3)+DT2*TERM1*C3(I)
      B(II,I3)=B(II,I3)-DT*TERM1*C3(I)
      A(JJ,J1)=A(JJ,J1)+DT2*TERM2*C1(I)
      B(JJ,J1)=B(JJ,J1)-DT*TERM2*C1(I)
      A(JJ,J2)=A(JJ,J2)+DT2*TERM2*C2(I)
      B(JJ,J2)=B(JJ,J2)-DT*TERM2*C2(I)
      A(JJ,J3)=A(JJ,J3)+DT2*TERM2*C3(I)
      B(JJ,J3)=B(JJ,J3)-DT*TERM2*C3(I)
      A(KK,K1)=A(KK,K1)+DT2*TERM3*C1(I)
      B(KK,K1)=B(KK,K1)-DT*TERM3*C1(I)
      A(KK,K2)=A(KK,K2)+DT2*TERM3*C2(I)
      B(KK,K2)=B(KK,K2)-DT*TERM3*C2(I)
      A(KK,K3)=A(KK,K3)+DT2*TERM3*C3(I)
    1 B(KK,K3)=B(KK,K3)-DT*TERM3*C3(I)
      RETURN
      END



      SUBROUTINE DMATRI(NOTRI,DUM,C1,C2,C3,SPH,D)
      INTEGER*2 NPTS
      COMMON /CM3/ NPTS(1300,3)
      DIMENSION DUM(1),C1(1),C2(1),C3(1),SPH(1),D(684,1)
      DATA EARTH/6.371E+06/
      DO 1 I=1,NOTRI
      II=NPTS(I,1)
      JJ=NPTS(I,2)
      KK=NPTS(I,3)
      CALL SEARCH(II,JJ,KK,I1,I2,I3,II)
      CALL SEARCH(II,JJ,KK,J1,J2,J3,JJ)
      CALL SEARCH(II,JJ,KK,K1,K2,K3,KK)
      FACT=1./(EARTH*120.)
      TA=SPH(II)*DUM(II)
      TB=SPH(JJ)*DUM(II)
      TC=SPH(KK)*DUM(II)
      TD=SPH(II)*DUM(JJ)
      TE=SPH(JJ)*DUM(JJ)
      TF=SPH(KK)*DUM(JJ)
      TG=SPH(II)*DUM(KK)
      TH=SPH(JJ)*DUM(KK)
      TI=SPH(KK)*DUM(KK)
      TERM1=6.*TA+2.*(TB+TC+TD+TE+TG+TI)+TF+TH
      TERM1=TERM1*FACT
      TERM2=6.*TE+2.*(TA+TB+TD+TF+TH+TI)+TC+TG
      TERM2=TERM2*FACT
      TERM3=6.*TI+2.*(TA+TC+TE+TF+TG+TH)+TB+TD
      TERM3=TERM3*FACT
      D(II,I1)=D(II,I1)+TERM1*C1(I)
      D(II,I2)=D(II,I2)+TERM2*C1(I)
      D(II,I3)=D(II,I3)+TERM3*C1(I)
      D(JJ,J1)=D(JJ,J1)+TERM1*C2(I)
      D(JJ,J2)=D(JJ,J2)+TERM2*C2(I)
```

```
      D(JJ,J3)=D(JJ,J3)+TERM3*C2(I)
      D(KK,K1)=D(KK,K1)+TERM1*C3(I)
      D(KK,K2)=D(KK,K2)+TERM2*C3(I)
    1 D(KK,K3)=D(KK,K3)+TERM3*C3(I)
      RETURN
      END



      SUBROUTINE AREAZ(NOTRI,OMEGA,NOPTS)
      INTEGER*2 NPTS
      COMMON /CM1/ AREA2(1296),A1(1296),A2(1296),A3(1296),
     1B1(1296),B2(1296),B3(1296)
      COMMON /CM2/ XLAT(684),XLON(684),XLON1(684)
      COMMON /CM3/ NPTS(1300,3)
      COMMON /CM4/ COSLAT(684),SINLAT(684)
      COMMON /CM6/ NTRI(46),NTRO(23)
      DATA RAD/6.283185308/,L/1/,LL/1/
      DO 6 I=1,NOPTS
    6 XLON(I)=RAD-XLON(I)
      DO 2 J=1,NOPTS
      IF(L.GT.23) GO TO 3
      IF(NTRO(L).NE.J) GO TO 3
      L=L+1
      XLON1(J)=XLON(J)-RAD
      GO TO 8
    3 XLON1(J)=XLON(J)
    8 COSLAT(J)=COS(XLAT(J))
      SINLAT(J)=SIN(XLAT(J))
    2 CONTINUE
      XLON1(1)=0.0
      XLON1(NOPTS)=0.0
      DO 1 I=1,NOTRI
      II=NPTS(I,1)
      JJ=NPTS(I,2)
      KK=NPTS(I,3)
      IF(LL.GT.46) GO TO 4
      IF(NTRI(LL).NE.I) GO TO 4
      LL=LL+1
      A1(I)=XLON1(KK)-XLON1(JJ)
      A2(I)=XLON1(II)-XLON1(KK)
      A3(I)=XLON1(JJ)-XLON1(II)
      GO TO 5
    4 A1(I)=XLON(KK)-XLON(JJ)
      A2(I)=XLON(II)-XLON(KK)
      A3(I)=XLON(JJ)-XLON(II)
    5 B1(I)=XLAT(JJ)-XLAT(KK)
      B2(I)=XLAT(KK)-XLAT(II)
    1 B3(I)=XLAT(II)-XLAT(JJ)
      NOTR=NOTRI-4
      DO 10 I=1,5
      A2(I)=0.0
   10 A3(I)=0.0
      DO 11 I=NOTR,NOTRI
      A1(I)=0.0
   11 A2(I)=0.0
      DO 12 I=1,NOTRI
   12 AREA2(I)=ABS(A1(I)*B2(I)-A2(I)*B1(I))
      DO 14 I=NOTR,NOTRI
   14 AREA2(I)=AREA2(1)
      DO 15 I=1,5
   15 A3(I)=-A1(I)
      DO 16 I=NOTR,NOTRI
   16 A1(I)=-A3(I)
      RETURN
      END
```

```
      SUBROUTINE SEARCH(I,J,K,INDEX1,INDEX2,INDEX3,II)
      INTEGER*2 NCOR
      COMMON /CM5/ NCOR(684,7)
      DIMENSION INDEX(3)
      JJ=I
      DO 1 M=1,3
      IF(M.EQ.3) JJ=K
      DO 2 L=1,7
      IF(NCOR(II,L).NE.JJ) GO TO 2
      INDEX(M)=L
      JJ=J
      GO TO 1
    2 CONTINUE
    1 CONTINUE
      INDEX1=INDEX(1)
      INDEX2=INDEX(2)
      INDEX3=INDEX(3)
      RETURN
      END



      SUBROUTINE AMATRI(NOTRI,A,DUM)
      INTEGER*2 NPTS
      COMMON /CM1/ AREA2(1296),A1(1296),A2(1296),A3(1296),
     1B1(1296),B2(1296),B3(1296)
      COMMON /CM3/ NPTS(1300,3)
      DIMENSION A(684,1),DUM(1)
      DATA EARTH/6.371E+06/
      DO 1 I=1,NOTRI
      II=NPTS(I,1)
      JJ=NPTS(I,2)
      KK=NPTS(I,3)
      CALL SEARCH(II,JJ,KK,I1,I2,I3,II)
      CALL SEARCH(II,JJ,KK,J1,J2,J3,JJ)
      CALL SEARCH(II,JJ,KK,K1,K2,K3,KK)
      FACT=AREA2(I)/120.
      TERM1=6.*DUM(II)+2.*DUM(JJ)+2.*DUM(KK)
      TERM2=2.*DUM(II)+2.*DUM(JJ)+DUM(KK)
      TERM3=2.*DUM(II)+DUM(JJ)+2.*DUM(KK)
      TERM4=2.*DUM(II)+6.*DUM(JJ)+2.*DUM(KK)
      TERM5=DUM(II)+2.*DUM(JJ)+2.*DUM(KK)
      TERM6=2.*DUM(II)+2.*DUM(JJ)+6.*DUM(KK)
      A(II,I1)=A(II,I1)+FACT*TERM1
      A(II,I2)=A(II,I2)+FACT*TERM2
      A(II,I3)=A(II,I3)+FACT*TERM3
      A(JJ,J1)=A(JJ,J1)+FACT*TERM2
      A(JJ,J2)=A(JJ,J2)+FACT*TERM4
      A(JJ,J3)=A(JJ,J3)+FACT*TERM5
      A(KK,K1)=A(KK,K1)+FACT*TERM3
      A(KK,K2)=A(KK,K2)+FACT*TERM5
    1 A(KK,K3)=A(KK,K3)+FACT*TERM6
      RETURN
      END



      SUBROUTINE CORIO(NOTRI,DUM,VECT,SPH1,SPH2)
      IMPLICIT REAL*8 (T)
      INTEGER*2 NPTS
      COMMON /CM1/ AREA2(1296),A1(1296),A2(1296),A3(1296),
     1B1(1296),B2(1296),B3(1296)
      COMMON /CM3/ NPTS(1300,3)
      DIMENSION DUM(1),VECT(1),SPH1(1),SPH2(1)
      DATA EARTH/6.371E+06/,OMEGA/7.292115432E-05/
      DO 1 I=1,NOTRI
      II=NPTS(I,1)
      JJ=NPTS(I,2)
```

```
      KK=NPTS(I,3)
      FACT=AREA2(I)*OMEGA/360.
      TAAA=DUM(II)*SPH1(II)*SPH2(II)
      TAAB=DUM(II)*SPH1(II)*SPH2(JJ)
      TAAC=DUM(II)*SPH1(II)*SPH2(KK)
      TABA=DUM(II)*SPH1(JJ)*SPH2(II)
      TABB=DUM(II)*SPH1(JJ)*SPH2(JJ)
      TABC=DUM(II)*SPH1(JJ)*SPH2(KK)
      TACA=DUM(II)*SPH1(KK)*SPH2(II)
      TACB=DUM(II)*SPH1(KK)*SPH2(JJ)
      TACC=DUM(II)*SPH1(KK)*SPH2(KK)
      TBAA=DUM(JJ)*SPH1(II)*SPH2(II)
      TBAB=DUM(JJ)*SPH1(II)*SPH2(JJ)
      TBAC=DUM(JJ)*SPH1(II)*SPH2(KK)
      TBBA=DUM(JJ)*SPH1(JJ)*SPH2(II)
      TBBB=DUM(JJ)*SPH1(JJ)*SPH2(JJ)
      TBBC=DUM(JJ)*SPH1(JJ)*SPH2(KK)
      TBCA=DUM(JJ)*SPH1(KK)*SPH2(II)
      TBCB=DUM(JJ)*SPH1(KK)*SPH2(JJ)
      TBCC=DUM(JJ)*SPH1(KK)*SPH2(KK)
      TCAA=DUM(KK)*SPH1(II)*SPH2(II)
      TCAB=DUM(KK)*SPH1(II)*SPH2(JJ)
      TCAC=DUM(KK)*SPH1(II)*SPH2(KK)
      TCBA=DUM(KK)*SPH1(JJ)*SPH2(II)
      TCBB=DUM(KK)*SPH1(JJ)*SPH2(JJ)
      TCBC=DUM(KK)*SPH1(JJ)*SPH2(KK)
      TCCA=DUM(KK)*SPH1(KK)*SPH2(II)
      TCCB=DUM(KK)*SPH1(KK)*SPH2(JJ)
      TCCC=DUM(KK)*SPH1(KK)*SPH2(KK)
      TERMA=6.*(TAAB+TAAC+TABA+TACA+TBAA+TBBB+TCAA+TCCC)
      TERMB=4.*(TABB+TACC+TBAB+TBBA+TCAC+TCCA)
      TERMC=2.*(TABC+TACB+TBAC+TBBC+TBCA+TBCB+TBCC)
      TERMC=TERMC+2.*(TCAB+TCBA+TCBB+TCBC+TCCB)
      TERMK=24.*TAAA+TERMA+TERMB+TERMC
      TERMK=TERMK*FACT
      TERMD=6.*(TAAA+TABB+TBAB+TBBA+TBBC+TBCB+TCBB+TCCC)
      TERME=4.*(TAAB+TABA+TBAA+TBCC+TCBC+TCCB)
      TERMF=2.*(TAAC+TABC+TACA+TACB+TACC+TBAC+TBCA)
      TERMF=TERMF+2.*(TCAA+TCAB+TCAC+TCBA+TCCA)
      TERMM=24.*TBBB+TERMD+TERME+TERMF
      TERMM=TERMM*FACT
      TERMG=6.*(TAAA+TACC+TBBB+TBCC+TCAC+TCBC+TCCA+TCCB)
      TERMH=4.*(TAAC+TACA+TBBC+TBCB+TCAA+TCBB)
      TERMI=2.*(TAAB+TABA+TABB+TABC+TACB+TBAA+TBAB+TBAC)
      TERMI=TERMI+2.*(TBBA+TBCA+TCAB+TCBA)
      TERMO=24.*TCCC+TERMG+TERMH+TERMI
      TERMO=TERMO*FACT
      VECT(II)=VECT(II)+TERMK
      VECT(JJ)=VECT(JJ)+TERMM
    1 VECT(KK)=VECT(KK)+TERMO
      RETURN
      END



      SUBROUTINE PGF(NOTRI,DUM,C1,C2,C3,SPH,F)
      IMPLICIT REAL*8 (T)
      INTEGER*2 NPTS
      COMMON /CM3/ NPTS(1300,3)
      DIMENSION DUM(1),C1(1),C2(1),C3(1),SPH(1),F(1)
      DATA EARTH/6.371E+06/
      DO 1 I=1,NOTRI
      II=NPTS(I,1)
      JJ=NPTS(I,2)
      KK=NPTS(I,3)
      FACT=1./(EARTH*24.)
      TERMA=DUM(II)*C1(I)*SPH(II)
      TERMB=DUM(II)*C1(I)*SPH(JJ)
      TERMC=DUM(II)*C1(I)*SPH(KK)
      TERMD=DUM(JJ)*C2(I)*SPH(II)
```

```
      TERME=DUM(JJ)*C2(I)*SPH(JJ)
      TERMF=DUM(JJ)*C2(I)*SPH(KK)
      TERMG=DUM(KK)*C3(I)*SPH(II)
      TERMH=DUM(KK)*C3(I)*SPH(JJ)
      TERMI=DUM(KK)*C3(I)*SPH(KK)
      TERMJ=2.*(TERMA+TERMD+TERMG)
      TERMK=TERMJ+TERMB+TERMC+TERME+TERMF+TERMH+TERMI
      TERMK=TERMK*FACT
      TERML=2.*(TERMB+TERME+TERMH)
      TERMM=TERML+TERMA+TERMC+TERMD+TERMF+TERMG+TERMI
      TERMM=TERMM*FACT
      TERMN=2.*(TERMC+TERMF+TERMI)
      TERMO=TERMN+TERMA+TERMB+TERMD+TERME+TERMG+TERMH
      TERMO=TERMO*FACT
      F(II)=F(II)+TERMK
      F(JJ)=F(JJ)+TERMM
    1 F(KK)=F(KK)+TERMO
      RETURN
      END



      SUBROUTINE TANVEC(NOTRI,DUM,SPH1,SPH2,G)
      IMPLICIT REAL*8 (T)
      INTEGER*2 NPTS
      COMMON /CM1/ AREA2(1296),A1(1296),A2(1296),A3(1296),
     1B1(1296),B2(1296),B3(1296)
      COMMON /CM3/ NPTS(1300,3)
      DIMENSION DUM(1),SPH1(1),SPH2(1),G(1)
      DATA EARTH/6.371E+06/
      DO 1 I=1,NOTRI
      II=NPTS(I,1)
      JJ=NPTS(I,2)
      KK=NPTS(I,3)
      FACT=AREA2(I)/(720.*EARTH)
      TAAA=DUM(II)*SPH1(II)*SPH2(II)
      TAAB=DUM(II)*SPH1(II)*SPH2(JJ)
      TAAC=DUM(II)*SPH1(II)*SPH2(KK)
      TABA=DUM(II)*SPH1(JJ)*SPH2(II)
      TABB=DUM(II)*SPH1(JJ)*SPH2(JJ)
      TABC=DUM(II)*SPH1(JJ)*SPH2(KK)
      TACA=DUM(II)*SPH1(KK)*SPH2(II)
      TACB=DUM(II)*SPH1(KK)*SPH2(JJ)
      TACC=DUM(II)*SPH1(KK)*SPH2(KK)
      TBAA=DUM(JJ)*SPH1(II)*SPH2(II)
      TBAB=DUM(JJ)*SPH1(II)*SPH2(JJ)
      TBAC=DUM(JJ)*SPH1(II)*SPH2(KK)
      TBBA=DUM(JJ)*SPH1(JJ)*SPH2(II)
      TBBB=DUM(JJ)*SPH1(JJ)*SPH2(JJ)
      TBBC=DUM(JJ)*SPH1(JJ)*SPH2(KK)
      TBCA=DUM(JJ)*SPH1(KK)*SPH2(II)
      TBCB=DUM(JJ)*SPH1(KK)*SPH2(JJ)
      TBCC=DUM(JJ)*SPH1(KK)*SPH2(KK)
      TCAA=DUM(KK)*SPH1(II)*SPH2(II)
      TCAB=DUM(KK)*SPH1(II)*SPH2(JJ)
      TCAC=DUM(KK)*SPH1(II)*SPH2(KK)
      TCBA=DUM(KK)*SPH1(JJ)*SPH2(II)
      TCBB=DUM(KK)*SPH1(JJ)*SPH2(JJ)
      TCBC=DUM(KK)*SPH1(JJ)*SPH2(KK)
      TCCA=DUM(KK)*SPH1(KK)*SPH2(II)
      TCCB=DUM(KK)*SPH1(KK)*SPH2(JJ)
      TCCC=DUM(KK)*SPH1(KK)*SPH2(KK)
      TERMA=6.*(TAAB+TAAC+TABA+TACA+TBAA+TBBB+TCAA+TCCC)
      TERMB=4.*(TABB+TACC+TBAB+TBBA+TCAC+TCCA)
      TERMC=2.*(TABC+TACB+TBAC+TBBC+TBCA+TBCB+TBCC)
      TERMC=TERMC+2.*(TCAB+TCBA+TCBB+TCBC+TCCB)
      TERMK=24.*TAAA+TERMA+TERMB+TERMC
      TERMK=TERMK*FACT
      TERMD=6.*(TAAA+TABB+TBAB+TBBA+TBBC+TBCB+TCBB+TCCC)
      TERME=4.*(TAAB+TABA+TBAA+TBCC+TCBC+TCCB)
```

```
      TERMF=2.*(TAAC+TABC+TACA+TACB+TACC+TBAC+TBCA)
      TERMF=TERMF+2.*(TCAA+TCAB+TCAC+TCBA+TCCA)
      TERMM=24.*TBBB+TERMD+TERME+TERMF
      TERMM=TERMM*FACT
      TERMG=6.*(TAAA+TACC+TBBB+TBCC+TCAC+TCBC+TCCA+TCCB)
      TERMH=4.*(TAAC+TACA+TBBC+TBCB+TCAA+TCBB)
      TERMI=2.*(TAAB+TABA+TABB+TABC+TACB+TBAA+TBAB+TBAC)
      TERMI=TERMI+2.*(TBBA+TBCA+TCAB+TCBA)
      TERMO=24.*TCCC+TERMG+TERMH+TERMI
      TERMO=TERMO*FACT
      G(II)=G(II)+TERMK
      G(JJ)=G(JJ)+TERMM
    1 G(KK)=G(KK)+TERMO
      RETURN
      END



      SUBROUTINE SOLUT(XLAT1,XLON1,UI,VI,PHII,NOPTS,WAVENO,
     1WAVVEL,AMP,KL,KK,KM,DT,TIME)
      DIMENSION XLAT1(1),XLON1(1),UI(1),VI(1),PHII(1)
      DATA EARTHA/6.371E+06/,OMEGA/7.292115432 5E-05/
      DATA BASEHT/5570./,GRAV/9.8/
      ASTAR=AMP*EARTHA
      WNP1=WAVENO+1.
      PHAZSP=WAVVEL/WAVENO
      B=(PHAZSP+2.*OMEGA/(WNP1*(WNP1+1.)))*(1./(1.-2./(WNP1*
     1(WNP1+1.))))
      A2=EARTHA**2
      DO 1 I=1,NOPTS
      TERM1=B/2.*(2.*OMEGA+B)*(COS(XLAT1(I))**2)
      TERM2A=.25*(ASTAR/(A2))**2
      TERM2B=COS(XLAT1(I))**(2*IFIX(WAVENO))
      IF(TERM2B.LT..1E-20) TERM2B=0.0
      TERM2C=WNP1*COS(XLAT1(I))**2
      TERM2D=2.*WAVENO**2-WAVENO-2.
      TERM2E=2.*WAVENO**2/(COS(XLAT1(I))**2)
      ATHETA=TERM1+TERM2A*TERM2B*(TERM2C+TERM2D-TERM2E)
      TERM1=2.*(OMEGA+B)*ASTAR/A2
      TERM2=WNP1*(WNP1+1.)
      TERM3=COS(XLAT1(I))**IFIX(WAVENO)
      TERM4=WAVENO**2+2.*WAVENO+2.
      TERM5=WNP1**2*COS(XLAT1(I))**2
      BTHETA=TERM1/TERM2*TERM3*(TERM4-TERM5)
      TERM1=.25*(ASTAR/(A2))**2
      TERM2=COS(XLAT1(I))**(2*IFIX(WAVENO))
      IF(TERM2.LT..1E-20) TERM2=0.0
      TERM3=WNP1*COS(XLAT1(I))**2-(WNP1+1.)
      CTHETA=TERM1*TERM2*TERM3
      TERM1=A2*ATHETA
      TERM2=A2*BTHETA*SIN(XLON1(I)*WAVENO-WAVVEL*TIME)
      TERM3=A2*CTHETA*(2.*SIN(XLON1(I)*WAVENO-WAVVEL*TIME)
     1**2-1.)
      PHII(I)=TERM1+TERM2+TERM3+BASEHT*GRAV
      TERM1=ASTAR*SIN(XLON1(I)*WAVENO-WAVVEL*TIME)
      TERM2=COS(XLAT1(I))**IFIX(WNP1)
      TERM3=WAVENO*ASTAR*SIN(WAVENO*XLON1(I)-WAVVEL*TIME)
      TERM4=COS(XLAT1(I))**IFIX(WAVENO-1.)
      TERM5=SIN(XLAT1(I))**2
      TERM6=B*A2*COS(XLAT1(I))
      UI(I)=-1./EARTHA*(TERM1*TERM2-TERM3*TERM4*TERM5-TERM6)
      TERM1=ASTAR*WAVENO*SIN(XLAT1(I))
      TERM2=COS(XLAT1(I))**IFIX(WAVENO-1.)
      TERM3=COS(XLON1(I)*WAVENO-WAVVEL*TIME)
    1 VI(I)=1./EARTHA*(TERM1*TERM2*TERM3)
      RETURN
      END
```

```
      SUBROUTINE EVAL(NOTRI,DUM,A,B,C)
      INTEGER*2 NPTS
      COMMON /CM2/ XLAT(684),XLON(684),XLON1(684)
      COMMON /CM3/ NPTS(1300,3)
      COMMON /CM6/ NTRI(46),NTRO(23)
      DIMENSION DUM(1),A(640),B(640),C(640)
      L=1
      NOTR=NOTRI/2
      DO 1 I=1,NOTR
      II=NPTS(I,1)
      JJ=NPTS(I,2)
      KK=NPTS(I,3)
      TA=DUM(II)-DUM(JJ)
      IF(NTRI(L).NE.I) GO TO 2
      TB=XLON1(II)-XLON1(KK)
      TD=XLON1(II)-XLON1(JJ)
      GO TO 3
    2 TB=XLON(II)-XLON(KK)
      TD=XLON(II)-XLON(JJ)
    3 TC=DUM(II)-DUM(KK)
      TF=XLAT(II)-XLAT(JJ)
      TG=XLAT(II)-XLAT(KK)
      C(I)=(TA*TB-TC*TD)/(TB*TF-TG*TD)
      IF(TD.EQ.0) GO TO 5
      B(I)=(TA-C(I)*TF)/TD
      GO TO 6
    5 B(I)=(TC-C(I)*TG)/TB
    6 IF(NTRI(L).NE.I) GO TO 4
      A(I)=DUM(II)-C(I)*XLAT(II)-B(I)*XLON1(II)
      L=L+1
      GO TO 1
    4 A(I)=DUM(II)-C(I)*XLAT(II)-B(I)*XLON(II)
    1 CONTINUE
      RETURN
      END



      SUBROUTINE DISPL(BFLD,NTABL,COA,COB,COC)
      INTEGER*2 NTABL
      DIMENSION BFLD(73,35),NTABL(72,18)
      DIMENSION COA(1),COB(1),COC(1)
      DATA RAD/6.283185308/,THETA/1.570796327/
      DRAD=RAD/72.
      DTHET=THETA-DRAD
      YLA=DTHET
      DO 1 J=1,18
      DO 2 I=1,72
      XLO=FLOAT(I-1)*DRAD
      TA=COA(NTABL(I,J))
      TB=COB(NTABL(I,J))
      TC=COC(NTABL(I,J))
      JJ=36-J
      BFLD(I,JJ)=TA+TB*XLO+TC*YLA
    2 CONTINUE
    1 YLA=YLA-DRAD
      DO 3 J=18,35
    3 BFLD(73,J)=BFLD(1,J)
      TD=-.2617993878
      TE=-.1745329252
      TF=-.0872664626
      BFLD(72,21)=COA(480)+COB(480)*TF-COC(480)*TD
      BFLD(72,20)=COA(559)+COB(559)*TF-COC(559)*TE
      BFLD(71,20)=COA(560)+COB(560)*TE-COC(560)*TE
      BFLD(72,19)=COA(638)+COB(638)*TF-COC(638)*TF
      BFLD(71,19)=COA(639)+COB(639)*TE-COC(639)*TF
      BFLD(72,18)=COA(638)+COB(638)*TF
      BFLD(71,18)=COA(640)+COB(640)*TE
      BFLD(70,18)=COA(640)+COB(640)*TD
      SUM=0.0
```

```
      DO 5 I=1,72
    5 SUM=SUM+BFLD(I,18)
      AVPHI=SUM/72.
      DO 4 J=1,17
      DO 4 I=1,73
    4 BFLD(I,J)=AVPHI
      RETURN
      END



      SUBROUTINE LONGOU (BFLD,FLD,XNOPO)
      DIMENSION FLD(63,63),BFLD(73,35)
      DO 10 I=1,63
      DO 10 J=1,63
      AJ=J-1.
      AI=I-1.
      CALL LLCVT3(AI,AJ,ALAT,ALON)
      BI=(350.0-ALON)/5.0+1.0
      IF(BI.LT.1.0) BI=(710.0-ALON)/5.0+1.0
      IF(ALAT.GT.84.99) ALAT=84.99
      BJ=(ALAT/5.0)+18.0
   10 CALL CINTRP(BI,BJ,BFLD,FLD(I,J),73,35)
      FLD(31,31)=XNOPO
      RETURN
      END



      SUBROUTINE CINTRP(FII,FJJ,F,AFF,K,L)
      DIMENSION F(K,L)
      I=FII
      J=FJJ
      AI=I
      BJ=J
      R=FII-AI
      S=FJJ-BJ
      IF(I.EQ.1) GO TO 10
      IF(I.EQ.K-1) GO TO 10
      IF(J.EQ.1) GO TO 10
      IF(J.EQ.L-1) GO TO 10
      A=((F(I,J+1)-F(I,J))+(F(I,J)-F(I,J-1)))*.5
      B=3.*(F(I,J+1)-F(I,J))-(2.*A+((F(I,J+2)-F(I,J+1))+(F(I
     1,J+1)-F(I,J)))*.5)
      C=(A+((F(I,J+2)-F(I,J+1))+(F(I,J+1)-F(I,J)))*.5)-2.*(F
     1(I,J+1)-F(I,J))
      F1=F(I,J)+S*(A+S*(B+S*C))
      I=I+1
      A=((F(I,J+1)-F(I,J))+(F(I,J)-F(I,J-1)))*.5
      B=3.*(F(I,J+1)-F(I,J))-(2.*A+((F(I,J+2)-F(I,J+1))+(F(I
     1,J+1)-F(I,J)))*.5)
      C=(A+((F(I,J+2)-F(I,J+1))+(F(I,J+1)-F(I,J)))*.5)-2.*(
     1F(I,J+1)-F(I,J))
      F2=F(I,J)+S*(A+S*(B+S*C))
      I=I+1
      A=((F(I,J+1)-F(I,J))+(F(I,J)-F(I,J-1)))*.5
      B=3.*(F(I,J+1)-F(I,J))-(2.*A+((F(I,J+2)-F(I,J+1))+(F(I
     1,J+1)-F(I,J)))*.5)
      C=(A+((F(I,J+2)-F(I,J+1))+(F(I,J+1)-F(I,J)))*.5)-2.*(F
     1(I,J+1)-F(I,J))
      F3=F(I,J)+S*(A+S*(B+S*C))
      I=I-3
      A=((F(I,J+1)-F(I,J))+(F(I,J)-F(I,J-1)))*.5
      B=3.*(F(I,J+1)-F(I,J))-(2.*A+((F(I,J+2)-F(I,J+1))+(F(I
     1,J+1)-F(I,J)))*.5)
      C=(A+((F(I,J+2)-F(I,J+1))+(F(I,J+1)-F(I,J)))*.5)-2.*(F
     1(I,J+1)-F(I,J))
      F4=F(I,J)+S*(A+S*(B+S*C))
```

```
      A=((F2-F1)+(F1-F4))*.5
      B=3.*(F2-F1)-(2.*A+((F3-F2)+(F2-F1))*.5)
      C=(A+((F3-F2)+(F2-F1))*.5)-2.*(F2-F1)
      AFF=F1+R*(A+R*(B+R*C))
      GO TO 20
   10 AFF=(1.-S)*((1.-R)*F(I,J)+R*F(I+1,J))+S*((1.-R)*F(I,J+
     11)+R*(F(I+1,J+1)))
   20 RETURN
      END



      SUBROUTINE LLCVT3(GVNI,GVNJ,FLAT,FLON)
      IF((GVNI.GT.30.9.AND.GVNI.LT.31.1).AND.(GVNJ.GT.30.9.A
     1ND.GVNJ.LT.31.1)) GO TO 1000
      SETJ=GVNJ-31.
      SETI=GVNI-31.
      STEST=0.
      ARTAN=ATAN2(SETJ,SETI)*57.29578
      ATEST=-10.0
      IF(SETJ.LT.STEST) GO TO 5
      SETK=350.0
      GO TO 10
    5 IF(ARTAN.GT.ATEST) GO TO 7
      SETK=ATEST
      GO TO 10
    7 SETK=350.0
   10 FLON=SETK-ARTAN
      SQJ=SETJ**2
      SQI=SETI**2
      RESQ=973.752025
      GROUP=(RESQ-SQI-SQJ)/(RESQ+SQI+SQJ)
      FLAT=ARSIN(GROUP)*57.29578
      GO TO 541
 1000 FLAT=90.0
      FLON=0.0
  541 RETURN
      END



      SUBROUTINE HARMAN(J1,JM,Z,Y,MM)
      DIMENSION FC(36,19),FS(36,19),Y(36),AF(19),BF(19),
     1PHASE(19),AMPL(19)
      IM=36
      I1   =   IM
      IF (J1) 10,10,27
   10 J1 = J1 + 1
      N11 = (I1/2) + 1
      N12 = N11 - 1
      E1 = 6.2831853/FLOAT(I1)
      E2 = 1.0/FLOAT(N12)
      WRITE(6,19) I1,JM,Z,N12
   19 FORMAT(//,'1','E-W GRID POINTS =',I3,5X,
     1'N-S GRID POINTS =',I3,5X,' GRID INCREMENT ='
     2,F5.1,5X,'HIGHEST WAVE NUMBER POSSIBLE =',I3,//)
      DO 20 I = 1,IM
      DO 20 J = 1,N11
      FC(I,J) = E2*COS(FLOAT((I   )*(J-1))*E1)
   20 FS(I,J) = E2*SIN(FLOAT((I   )*(J-1))*E1)
      DO 25 I = 1,IM
      FC(I,1) = 0.5*FC(I,1)
   25 FC(I,N11) = 0.5*FC(I,N11)
   27 N11 = (I1/2) + 1
      N12 = N11 - 1
      DO 40 J = 1,N11
      SUM = 0.0
      DO 30 I = 1,IM
```

```
 30 SUM = SUM + Y(I)*FC(I,J)
 40 AF(J) = SUM
    BF(1) = 0.0
    BF(N11) = 0.0
    DO 60 J = 2,N12
    SUM = 0.0
    DO 50 I = 1,IM
 50 SUM = SUM + Y(I)*FS(I,J)
 60 BF(J) = SUM
    DO 75 J = 1,N11
    PHASE(J) = 0.0
    IF (BF(J).EQ.0.0) GO TO 75
    IF ((BF(J).NE.0.0).AND.(AF(J).EQ.0.0)) GO TO 70
    PHASE(J) = ATAN(BF(J)/AF(J))
    GO TO 75
 70 PHASE(J) = 90.0*(BF(J)/ABS(BF(J)))
 75 CONTINUE
    DO 95 J = 1,N11
    IF (ABS(PHASE(J)) - 0.8) 80,80,90
 80 AMPL(J) = AF(J)/COS(PHASE(J))
    GO TO 95
 90 AMPL(J) = BF(J)/SIN(PHASE(J))
 95 CONTINUE
    DO 100 J = 1,N11
100 PHASE(J) = 57.29578*PHASE(J)
    DO 120 J = 2,N11
    IF (AMPL(J)) 110,120,120
110 IF (PHASE(J)) 112,112,115
112 PHASE(J) = PHASE(J) + 180
    GO TO 118
115 PHASE(J) = PHASE(J) - 180
118 AMPL(J) = -AMPL(J)
120 CONTINUE
    WRITE(6,121)
121 FORMAT( /,' ',T5,'LAT CIR',T20,'ARITHMETIC MEAN',T40,
   1'AMPLITUDE',T60,'PHASE')
    WRITE(6,122) MM,AMPL(1), ((AMPL(I), PHASE(I)),I=2,N11)
122 FORMAT(T6,I2,T16,F11.2,5(T36,F8.2,12X,F8.2,/))
    RETURN
    END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                       C
C                                                       C
C    THE PURPOSE OF THIS PROGRAM IS TO TAKE A N X N     C
C     MATRIX AND REDUCE IT TO A N X 7 MATRIX TO CONSERVE C
C     CORE REQUIREMENTS                                 C
C                                                       C
C                                                       C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                       C
C                                                       C
C    NPTS= THE GLOBAL CORRESPONDENCE TABLE              C
C    NCOR= THE GLOBAL CORRELATION TABLE. THIS IS THE MATRIXC
C     THAT THIS PROGRAM WILL GENERATE. IT CORRELATES THE C
C     GLOBAL CORRESPONDENS NUMBER OF THE N X N MATRIX   C
C     WITH IT'S POSITION IN THE N X 7 MATRIX.           C
C    NTRI5= THE VECTOR THAT CONTAINS THE NUMBER OF THE  C
C     POINT THAT IS SUPPORTED BY ONLY FIVE TRIANGLES VICE C
C     SIX TRIANGLES.                                    C
C    NX= A TEMPORY HOLDING VECTOR                       C
C                                                       C
C                                                       C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      INTEGER*2 NPTS,NCOR,NTRI5,NX
      COMMON /CM1/ NCOR(642,7),NPTS(1280,3)
      DIMENSION NTRI5(12),ISAVE(6),NX(7)
      DATA N/ 8/,IKOUNT/1/
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                       C
C                                                       C
C    NOTRI= THE NUMBER OF TRIANGLES GIVEN 8 SEGMENTS PER C
C     MAJOR SPHERICAL TRIANGLE'S SIDE                   C
C                                                       C
C                                                       C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      NOTRI=1280
      DO 1 I=1,NOTRI,5
      L=I+4
    1 READ(5,100)((NPTS(K,J),J=1,3),K=I,L)
  100 FORMAT(5(3I5))
      READ(5,101) NTRI5
  101 FORMAT(10I8)
      NOPTS=642
      DO 20 I=1,NOPTS
      DO 20 J=1,7
   20 NCOR(I,J)=0
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                       C
C                                                       C
C    THE ENTIRE 2 DO LOOP COMPUTES THE GLOBAL CORRELATION C
C     TABLE. THE DO 2 LOOP SEARCHES EVERY POINT TO FIND C
C     THE FIVE OR SIX TRIANGLES THAT SUPPORT THAT POINT. C
C     SUBROUTINE CHECK DETERMINES IF THE POINT LIES IN  C
C     THE TRIANGLE. IF THE POINT LIES IN THE TRIANGLE THEN C
C     THE TRIANGLE NUMBER IS STORED IN THE ISAVE VECTOR. C
C     THIS IS DONE UNTIL THE PROPER NUMBER OF TRIANGLES C
C     HAVE BEEN FOUND. ASSUMING THAT A POINT IS SUPPORTED C
C     BY SIX TRIANGLES THE NEXT STEP IS TO SORT OUT THE C
C     SEVEN DIFFERENT NUMBERS FROM THE 18 AVAILABLE.    C
C     SUBROUTINE SORT ACCOMPLISHES THIS AND THEN ARRANGES C
C     THE SEVEN NUMBERS IN ASCENDING ORDER. THESE SEVEN C
C     NUMBERS ARE THEN PLACED IN THE POINT'S ROW OF THE C
C     GLOBAL CORRELATION TABLE.                         C
C                                                       C
C                                                       C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      DO 2 L=1,NOPTS
      LL=6
      IF(NTRI5(IKOUNT).NE.L) GO TO 3
      LL=5
      IKOUNT=IKOUNT+1
```

```
    3 ICOUNT=0
      DO 4 J=1,NOTRI
      IFLAG=0
      CALL CHECK(NPTS(J,1),NPTS(J,2),NPTS(J,3),L,IFLAG)
      IF(IFLAG.EQ.0) GO TO 4
      ICOUNT=ICOUNT+1
      ISAVE(ICOUNT)=J
      IF(ICOUNT.EQ.LL) GO TO 5
    4 CONTINUE
    5 CALL SORT(ISAVE,LL,NX)
      IF(LL.EQ.5) LLL=6
      IF(LL.EQ.6) LLL=7
      DO 6 I=1,LLL
    6 NCOR(L,I)=NX(I)
    2 CONTINUE
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                     C
C                                                                     C
C    THIS SECTION OUTPUTS THE THE GLOBAL CORRELATION TABLE.           C
C     SINCE THE TABLE NEED BY COMPUTED ONLY ONCE, IT CAN              C
C     BE COMPUTED IN A SEPARATE PROGRAM AND OUTPUT ON                 C
C     CARDS AND READ IN AS DATA IN THE MAIN PROGRAM.                  C
C                                                                     C
C                                                                     C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      WRITE(6,102)((NCOR(K,J),J=1,7),K=1,NOPTS)
  102 FORMAT(/1X,7I10)
      DO 21 I=1,NOPTS
   21 WRITE(7,103)(NCOR(I,J),J=1,7)
  103 FORMAT(7I10)
      STOP
      END



      SUBROUTINE CHECK(DUM1,DUM2,DUM3,L,IFLAG)
      INTEGER*2 DUM1,DUM2,DUM3
      IF(DUM1.EQ.L) IFLAG=1
      IF(DUM2.EQ.L) IFLAG=1
      IF(DUM3.EQ.L) IFLAG=1
      RETURN
      END



      SUBROUTINE SORT(ISAVE,LL,NX)
      INTEGER*2 NPTS,NCOR,NTRI5,NX
      COMMON /CM1/ NCOR(642,7),NPTS(1280,3)
      DIMENSION NX(7),ISAVE(6)
      NX(1)=NPTS(ISAVE(1),1)
      NX(2)=NPTS(ISAVE(1),2)
      NX(3)=NPTS(ISAVE(1),3)
      L=3
    6 DO 1 I=2,LL
    5 DO 3 K=1,3
      DO 2 J=1,L
      IF(NPTS(ISAVE(I),K).EQ.NX(J)) GO TO 3
      IF(J.NE.L) GO TO 2
      NX(L+1)=NPTS(ISAVE(I),K)
      GO TO 4
    2 CONTINUE
    3 CONTINUE
      GO TO 1
    4 L=L+1
      IF(K.EQ.3) GO TO 1
      GO TO 5
    1 CONTINUE
      IF(LL.EQ.5) LLL=6
```

```
      IF(LL.EQ.6) LLL=7
      DO 8 J=1,7
      DO 7 I=2,LLL
      IF(NX(I-1).LT.NX(I)) GO TO 7
      ISWAP=NX(I-1)
      NX(I-1)=NX(I)
      NX(I)=ISWAP
    7 CONTINUE
    8 CONTINUE
      RETURN
      END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      C
C                                                                      C
C     PROGRAM TO GENERATE GEODESIC GRID                                C
C  SPRLON= LONGITUDE OF MAJOR SPHERICAL TRIANGLES                      C
C  SPRLAT = LATITUDE OF MAJOR SPHERICAL TRIANGLES                      C
C                                                                      C
C                                                                      C
C     NPTS(5120,3)= THE GLOBAL CORRESPONDENCE TABLE FOR                C
C      5120 TRIANGLES, N= 16.                                          C
C     XLAT(642)= THE LATITUDE VECTOR FOR N= 8                          C
C     XLON1(642) THE LONGITUDE VECTOR FOR N= 8                         C
C                                                                      C
C                                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      INTEGER*2 NPTS(5120,3)
      DIMENSION SPRLAT(12),SPRLON(12),XLAT(81,49)
      DIMENSION XLON(81,49)
      DIMENSION XLAT1(642),XLON1(642)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      C
C                                                                      C
C     THESE FUNCTION STATEMENTS ARE FOR THE LAW OF SINES               C
C      (SINLAW), THE LAW OF COSINES FOR SIDES (COSLAS), AND            C
C      THE LAW OF COSINES FOR ANGLES (COSLAA). ALL THREE               C
C      FUNCTION STATEMENTS ARE FOR SPHERICAL TRIANGLES.                C
C                                                                      C
C                                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      SINLAW(A,ANGLA,B)=ARSIN(SIN(B)*SIN(ANGLA)/SIN(A))
      COSLAS(C,A,ANGLB)=ARCOS(COS(C)*COS(A)+
     1SIN(C)*SIN(A)*COS(ANGLB))
      COSLAA(ANGLA,ANGLB,ANGLC)=ARCOS((COS(ANGLB)+
     1COS(ANGLC)*COS(ANGLA))
     1/(SIN(ANGLC)*SIN(ANGLA)))
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      C
C                                                                      C
C     ANGLST= THE INTERIOR ANGLE OF A MAJOR SPHERICAL                  C
C      TRIANGLE, 72 DEGREES                                            C
C     XNOPO= THE LATITUDE OF THE NORTH POLE                            C
C                                                                      C
C                                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      DATA ANGLST/1.256637062/,N/ 8/,XNOPO/1.570796327/
      DATA RAD/57.29577951/
      NTP1=3*N+1
      NFP1=5*N+1
      NP1=N+1
      NT=2*N
      NP2=N+2
      NM1=N-1
      NM2=N-2
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      C
C                                                                      C
C     NOTRI= NUMBER OF TRIANGLES OVER THE GLOBE WHERE N=               C
C      THE NUMBER OF SEGMENTS EACH MAJOR SPHERICAL TRIANGLE            C
C      'S SIDES ARE SUBDIVIDED INTO.                                   C
C     NOPTS= THE NUMBER OF POINTS OVER THE GLOBE                       C
C                                                                      C
C                                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      NOTRI=20*N**2
      NOPTS=10*N**2+2
      DO 12 I=1,NTP1
      DO 12 J=1,NFP1
      XLON(J,I)=0.0
   12 XLAT(J,I)=0.0
      READ(5,100) SPRLON
      READ(5,100) SPRLAT
  100 FORMAT(6F12.7)
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                       C
C                                                       C
C    SETS UP MODEL PTS. WHERE EACH LONGITUDE LEG IS DIVID- C
C      ED INTO N SEGMENTS AND THE PTS. ARE JOINED BY GREAT C
C      CIRCES                                           C
C                                                       C
C                                                       C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      XLON(1,NP1)=SPRLON(2)/RAD
      XLON(1,1)=SPRLON(1)/RAD
      XLON(NP1,NP1)=SPRLON(3)/RAD
      XLAT(1,NP1)=SPRLAT(2)/RAD
      XLAT(1,1)=SPRLAT(1)/RAD
      XLAT(NP1,NP1)=SPRLAT(3)/RAD
      ARCLEN=COSLAA(ANGLST,ANGLST,ANGLST)
      SEG=ARCLEN/FLOAT(N)
      DO 1 I=2,N
      XLON(1,I)=SPRLON(1)/RAD
      XLON(I,I)=SPRLON(3)/RAD
      COLAT=FLOAT(I-1)*SEG
      XLAT(1,I)=(XNOPO-COLAT)
    1 XLAT(I,I)=XLAT(1,I)
      DO 2 I=2,N
      K=I-1
      ARCLEN=COSLAS(FLOAT(I)*SEG,FLOAT(I)*SEG,ANGLST)
      ANGLC=SINLAW(ARCLEN,ANGLST,FLOAT(I)*SEG)
      ARCLEN=ARCLEN/FLOAT(I)
      ANGLA=ANGLST/FLOAT(I)
      DO 2 J=1,K
      XLON(J+1,I+1)=ANGLA*FLOAT(J)
      COLAT=COSLAS(FLOAT(I)*SEG,ARCLEN*FLOAT(J),ANGLC)
    2 XLAT(J+1,I+1)=XNOPO-COLAT
      XLON(1,NTP1)=XLON(1,1)
      XLAT(1,NTP1)=-XLAT(1,1)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                       C
C                                                       C
C    THIS DO LOOP MIRRORS THE SOLUTION OF THE FIRST     C
C     MAJOR SPHERICAL NORTHERN HEMISPHERIC TRIANGLE     C
C     INTO THE AJOINING FOUR NORTHERN HEMISPHERIC       C
C     MAJOR SPHERICAL TRIANGLES.                        C
C                                                       C
C                                                       C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      DO 3 K=1,4
      DO 3 I=2,NP1
      DO 3 J=2,I
      XLAT(K*I+J-K,I)=XLAT(J,I)
    3 XLON(K*I+J-K,I)=XLON(J,I)+FLOAT(K)*ANGLST
      M=3*N
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                       C
C                                                       C
C    THIS DO LOOP MIRRORS THE NORTHERN HEMISPHERIC SOLUTIONC
C     INTO THE SOUTHERN HEMISPERE.                      C
C                                                       C
C                                                       C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      DO 4 I=2,NP1
      II=(I-1)*5+1
      DO 14 J=1,II
      XLAT(J,M)=-XLAT(J,I)
   14 XLON(J,M)=XLON(J,I)+ANGLST/2.
    4 M=M-1
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                      C
C                                                      C
C   THIS NEXT SEQUENCE SOLVES THE EQUATORIAL MAJOR     C
C    SPHERICAL TRIANGLES                               C
C                                                      C
C                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      A=XNOPO-XLAT(1,NP1)
      C=XNOPO-XLAT(1,NT+1)
      ANGLB=ANGLST/2.
      ARCLEN=COSLAS(C,A,ANGLB)
      ARCLEN=ARCLEN/FLOAT(N)
      M=N
      DO 5 I=1,NM1
      COLAT=COSLAS(A,ARCLEN*FLOAT(I),ANGLST*2.)
      XLAT(1,NP1+I)=XNOPO-COLAT
      XLON(1,NP1+I)=SINLAW(COLAT,ANGLST*2.,ARCLEN*FLOAT(I))
      XLAT(M,NP1+I)=XLAT(1,NP1+I)
      XLON(M,NP1+I)=ANGLST-XLON(1,NP1+I)
    5 M=M-1
      M=N
      DO 6 I=1,NM2
      MM1=M-1
      A=XNOPO-XLAT(1,NP1+I)
      B=XNOPO-XLAT(M,NP1+I)
      ANGLB=XLON(M,NP1+I)-XLON(1,NP1+I)
      ARCLEN=COSLAS(A,B,ANGLB)
      ANGLC=SINLAW(ARCLEN,ANGLB,B)
      IF(I.LE.N/2) GO TO 16
      ANGLC=(XNOPO-ANGLC)+XNOPO
   16 ARCLEN=ARCLEN/FLOAT(MM1)
      DO 7 J=2,MM1
      COLAT=COSLAS(A,ARCLEN*FLOAT(J-1),ANGLC)
      XLAT(J,NP1+I)=XNOPO-COLAT
    7 XLON(J,NP1+I)=SINLAW(COLAT,ANGLC,ARCLEN*FLOAT(J-1))+
     1XLON(1,NP1+I)
    6 M=M-1
      M=N
      DO 8 I=1,NM1
      XLAT(NP1,NP1+I)=XLAT(1,NP1+I)
    8 XLON(NP1,NP1+I)=XLON(1,NP1+I)+ANGLST
      DO 9 I=2,NM1
      ANGLA=XLON(NP1,NP1+I)-XLON(NP1-I,NP1+I)
      C=XNOPO-XLAT(NP1-I,NP1+I)
      B=XNOPO-XLAT(NP1,NP1+I)
      ARCLEN=COSLAS(B,C,ANGLA)
      ANGLB=SINLAW(ARCLEN,ANGLA,C)
      IF(I.LE.N/2) GO TO 17
      ANGLB=(XNOPO-ANGLB)+XNOPO
   17 ARCLEN=ARCLEN/FLOAT(I)
      DO 9 J=2,I
      COLAT=COSLAS(B,ARCLEN*FLOAT(J-1),ANGLB)
      XLAT(NP2-J,NP1+I)=XNOPO-COLAT
    9 XLON(NP2-J,NP1+I)=XLON(NP1,NP1+I)-
     1SINLAW(COLAT,ANGLB,ARCLEN*FLOAT(J-1))
      DO 10 K=1,4
      DO 10 I=1,N
      DO 10 J=1,NM1
      XLAT(K*N+1+I,NP1+J)=XLAT(I+1,NP1+J)
   10 XLON(K*N+1+I,NP1+J)=XLON(I+1,NP1+J)+FLOAT(K)*ANGLST
      WRITE(6,104) N,NOPTS
  104 FORMAT(//1X,'THE NUMBER OF PTS IN THE GEODESIC GRID'
     1' DIVIDED INTO',2X,I2,2X,'SEGMENTS IS',2X,I4)
     12X,I2,2X,'SEGMENTS IS',2X,I4)
  105 FORMAT(1X,/24F5.0)
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                    C
C                                                                    C
C   THE NEXT SECTION WRITES THE TWO DIMENSIONAL MATRIX               C
C    CONTAINING THE LATITUDES AND THE LONGITUDES INTO A              C
C    ONE DIMENSIONAL VECTOR, ONE FOR THE LATITUDE AND ONE            C
C    FOR THE LONGITUDE.                                              C
C                                                                    C
C                                                                    C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      L=2
      DO 20 J=2,NP1
      MM=(J-1)*5
      DO 20 I=1,MM
      XLAT1(L)=XLAT(I,J)
      XLON1(L)=XLON(I,J)
   20 L=L+1
      NTT=NT+1
      DO 41 J=NP2,NTT
      K=5*N
      DO 41 I=1,K
      XLAT1(L)=XLAT(I,J)
      XLON1(L)=XLON(I,J)
   41 L=L+1
      NTT=NTT+1
      KK=NTP1-1
      DO 21 J=NTT,KK
      K=K-5
      DO 21 I=1,K
      XLAT1(L)=XLAT(I,J)
      XLON1(L)=XLON(I,J)
   21 L=L+1
      XLAT1(1)=XLAT(1,1)
      XLON1(1)=XLON(1,1)
      XLAT1(NOPTS)=XLAT(1,NTP1)
      XLON1(NOPTS)=XLON(1,NTP1)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                    C
C                                                                    C
C   THIS SECTION DEFINES THE GLOBAL CORRESPONDENCE NUMBER            C
C    FOR EACH POINT IN THE VECTORS CONTAINING THE                    C
C    LATITUDES AND THE LONGITUDES.                                   C
C                                                                    C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      DO 22 I=1,5
      NPTS(I,1)=1
      NPTS(I,2)=I+1
   22 NPTS(I,3)=I+2
      NPTS(5,3)=2
      L=6
      II=2
      ISAVE1=II
      III=7
      ICOUNT=2
      ISAVE=7
      DO 23 J=3,NP1
      DO 25 K=1,5
      DO 24 I=1,ICOUNT
      NPTS(L,1)=II
      NPTS(L,2)=III
      NPTS(L,3)=III+1
      IF(K.EQ.5.AND.I.EQ.ICOUNT) GO TO 44
      IF(I.EQ.ICOUNT) GO TO 24
      NPTS(L+1,1)=II
      NPTS(L+1,2)=III+1
      NPTS(L+1,3)=II+1
      IF(K.EQ.5.AND.I.EQ.ICOUNT-1) NPTS(L+1,3)=ISAVE1
      II=II+1
      III=III+1
      L=L+2
      GO TO 24
   44 NPTS(L,1)=ISAVE1
```

```
      NPTS(L,3)=ISAVE
   24 CONTINUE
      L=L+1
   25 III=III+1
      II=ISAVE
      ISAVE1=II
      III=ISAVE+(J-1)*5
      ISAVE=III
   23 ICOUNT=ICOUNT+1
      III=5*N+II
      ISAVE=III
      M=NFP1-1
      MT=NT+1
      DO 27 J=NP2,MT
      DO 26 I=1,M
      NPTS(L,1)=II
      NPTS(L,2)=III
      NPTS(L,3)=II+1
      NPTS(L+1,1)=II+1
      NPTS(L+1,2)=III
      NPTS(L+1,3)=III+1
      IF(I.NE.M) GO TO 45
      NPTS(L,3)=ISAVE1
      NPTS(L+1,1)=ISAVE1
      NPTS(L+1,3)=ISAVE
   45 II=II+1
      III=III+1
      L=L+2
   26 CONTINUE
      II=ISAVE
      III=ISAVE+5*N
      ISAVE=III
   27 ISAVE1=II
      MT=MT+1
      MMT=NTP1-1
      ICOUNT=ICOUNT-1
      ISAVE=III
      KKK=NM1
      DO 28 J=MT,MMT
      DO 29 K=1,5
      DO 30 I=1,ICOUNT
      NPTS(L,1)=II
      NPTS(L,2)=III
      NPTS(L,3)=II+1
      IF(K.EQ.5.AND.I.EQ.ICOUNT) GO TO 46
      IF(I.EQ.ICOUNT) GO TO 30
      NPTS(L+1,1)=II+1
      NPTS(L+1,2)=III
      NPTS(L+1,3)=III+1
      IF(K.EQ.5.AND.I.EQ.ICOUNT-1) NPTS(L+1,3)=ISAVE
      II=II+1
      III=III+1
      L=L+2
      GO TO 30
   46 NPTS(L,1)=ISAVE1
      NPTS(L,2)=ISAVE-1
   30 CONTINUE
      L=L+1
   29 II=II+1
      II=ISAVE
      III=ISAVE+KKK*5
      ISAVE1=II
      KKK=KKK-1
      ISAVE=III
   28 ICOUNT=ICOUNT-1
      DO 31 I=1,5
      NPTS(NOTRI-5+I,1)=NOPTS-6+I
      NPTS(NOTRI-5+I,2)=NOPTS
   31 NPTS(NOTRI-5+I,3)=NOPTS-5+I
      NPTS(NOTRI,3)=NOPTS-5
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                       C
```

```
C                                                                       C
C    THE LAST SECTION OUTPUTS THE GLOBAL CORRESPONDENCE                 C
C     TABLE, THE LATITUDE AND THE LONGITUDE VECTORS                     C
C                                                                       C
C                                                                       C
 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      K=NOTRI/6
      DO 32 I=1,K
      II=K+I
      III=2*K+I
      IIII=3*K+I
      IIIII=4*K+I
      IIIIII=5*K+I
   32 WRITE(6,110) I,(NPTS(I,J),J=1,3),II,(NPTS(II,JJ),JJ=1,
     13),III,(NPTS(III,JJJ),JJJ=1,3),IIII,(NPTS(IIII,JJJJ),J
     2JJJ=1,3),IIIII3(NPTS(IIIII,JJJJJ),JJJJJ=1,3),IIIIII,(N
     3PTS(IIIIII,JJJJJJ),JJJJJJ=1,3)
  110 FORMAT(/1X,6(4I5))
      WRITE(7,112) XLAT1
      WRITE(7,112) XLON1
  112 FORMAT(8F10.6)
      STOP
      END
```

LIST OF REFERENCES

1. Archer, D. A., The Finite Element Method, unpublished paper presented at the Naval Postgraduate School, Monterey, California, 11 August 1975.

2. Aziz, A. K. (ed.), The Mathematical Foundation of the Finite Element Method with Applications to Partial Differential Equations, Academic Press, 1972.

3. Cullen, M. J. P., "Integrations of the Primitive Equations on a Sphere Using the Finite Element Method," Quarterly Journal of the Royal Meteorological Society, v. 100, p. 555-562, 1974.

4. Desai, C. S., Introduction to the Finite Element Method A Numerical Method for Engineering Analysis, Van Nostrand Reinhold Company, 1972.

5. George, J. Alan, Computer Implementation of the Finite Element Method, Ph.D. Thesis, Stanford University, February, 1971.

6. Haltiner, G. J., and Martin, F. L., Dynamical and Physical Meteorology, p. 52-53, McGraw-Hill, 1957.

7. Haurwitz, B., "The Motion of Atmospheric Disturbances on the Spherical Earth." J. Marine Research (Sears Foundation), 3, p. 254-267, 1940.

8. Leslie, L. M., "Comparative Test of Direct and Iterative Methods for Solving Helmholtz Type Equations," Monthly Weather Review, v. 101, no. 3, p. 235-239, March 1973.

9. Maher, D. E., Experiments with a 5-Level Global Primitive Equation Atmospheric Model Using Analytically Determined Fields, M. S. Thesis, Naval Postgraduate School, Monterey, California, 1974.

10. Martin, H. C., and Carey, G. F., Introduction to Finite Element Analysis, McGraw-Hill, 1973.

11. Monaco, A. V., An Atmospheric Global Prediction Model Using a Modified Arakawa Differencing Scheme, M. S. Thesis, Naval Postgraduate School, Monterey, California, 1975.

12. Norrie, D. H., and de Vries, G., The Finite Element Method, Academic Press, 1973.

13. Phillips, N. A., "Numerical Integration of the Primitive Equations on the Hemisphere," Monthly Weather Review, p. 333-345, 1959.

14. Sadourny, Arakawa, and Mintz, Integration of the Non-Divergent Barotropic Vorticity Equation with an Icosahedral-Hexagonal Grid for the Sphere, Department of Meteorology, University of California, Los Angeles, Technical Report No. 2, November 1967.

15. Schultz, M. H., Spline Analysis, Prentice-Hall, Inc., 1973.

16. Smith, G. N., An Introduction to Matrix and Finite Element Methods in Civil Engineering, Applied Science Publishers, Ltd., 1971.

17. Strang, G., and Fix, G. J., An Analysis of the Finite Element Method, Prentice Hall, Inc., 1973.

18. Williamson, D. R., "Integration of the Barotropic Vorticity Equation on a Spherical Geodesic Grid," Tellus xx, v. 20, no. 4, p. 642-653, November 1968.

19. Ural, O., Finite Element Method, Basic Concepts and Applications, Intext Educational Publishers, 1973.

20. Zienkiewicz, O. C., The Finite Element Method in Engineering Science, p. 116-117, McGraw-Hill, 1971.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Commanding Officer Naval Weather Service Command Headquarters 3101 Building 200 Washington Navy Yard Washington, D. C. 20374	1
4.	Dean of Research Naval Postgraduate School Monterey, California 93940	2
5.	Naval Oceanographic Office Library (Code 3330) Washington, D. C. 20373	1
6.	Commanding Officer Fleet Numerical Weather Central Naval Postgraduate School Monterey, California 93940	2
7.	Officer in Charge Environmental Prediction Research Facility Naval Postgraduate School Monterey, California 93940	2
8.	Commander, Air Weather Service Military Airlift Command United States Air Force Scott Air Force Base, Illinois 62226	1
9.	AFCRL - Research Library L. G. Hanscom Field ATTN: Nancy Davis/Stop 29 Bedford, Massachusetts 01730	1
10.	Dr. G. J. Haltiner, Code 51Ha Chairman, Department of Meteorology Naval Postgraduate School Monterey, California 93940	5

11. Dr. R. T. Williams, Code 51Wu — 5
Department of Meteorology
Naval Postgraduate School
Monterey, California 93940

12. Dr. R. L. Alberty — 1
National Severe Storms Laboratory
University of Oklahoma
Norman, Oklahoma 73069

13. Dr. R. Alexander — 1
The RAND Corporation
1700 Main Street
Santa Monica, California 90406

14. Dr. A. Arakawa — 1
Department of Meteorology
University of California
Los Angeles, California 90024

15. Dr. David Archer — 2
Mathematics Department
University of North Carolina - Charlotte
University of North Carolina Station
Charlotte, North Carolina 28223

16. Atmospheric Sciences Library — 1
National Oceanic and Atmospheric Administration
Silver Spring, Maryland 20910

17. Mr. E. Barker — 1
Environmental Prediction Research Facility
Naval Postgraduate School
Monterey, California 93940

18. Dr. F. P. Bretherton — 1
National Center for Atmospheric Research
P.O. Box 3000
Boulder, Colorado 80303

19. Dr. John Brown — 1
National Meteorological Center/NOAA
World Weather Building
Washington, D. C. 20233

20. Dr. Gilles Cantin — 1
Department of Mechanical Engineering
Naval Postgraduate School
Monterey, California 93940

21. Dr. C. P. Chang, Code 51Cj — 1
Department of Meteorology
Naval Postgraduate School
Monterey, California 93940

22. Professor J. G. Charney — 1
54-1424
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

23. Dr. C. Comstock, Code 53Zk — 1
Department of Mathematics
Naval Postgraduate School
Monterey, California 93940

24. Mr. M. J. P. Cullen — 1
Meteorological Office
London Road
Bracknell Berkshire RG122SZ
England

25. Dr. D. Dietrich — 1
Code 7750
Naval Research Laboratory
Washington, D. C. 20375

26. Dr. R. L. Elsberry, Code 51Es — 1
Department of Meteorology
Naval Postgraduate School
Monterey, California 93940

27. Dr. Frank D. Faulkner — 1
Department of Mathematics
Naval Postgraduate School
Monterey, California 93940

28. Dr. J. A. Galt — 1
NOAA - Pac. Mar. Envir. Lab.
University of Washington WB-10
Seattle, Washington 98105

29. Dr. W. L. Gates — 1
The RAND Corporation
1700 Main Street
Santa Monica, California 90406

30. Dr. Earl Gossard — 1
Wave Propagation Laboratory
NOAA/ERL
Boulder, Colorado 80302

31. Dr. R. L. Haney, Code 51Hy — 1
Department of Meteorology
Naval Postgraduate School
Monterey, California 93940

32. Lieutenant Donald E. Hinsman — 2
Fleet Numerical Weather Central
Naval Postgraduate School
Monterey, California 93940

33. Dr. J. Holton — 1
Department of Atmospheric Sciences
University of Washington
Seattle, Washington 98105

34. Dr. B. J. Hoskins — 1
Department of Geophysics
University of Reading
Reading
United Kingdom

35. Dr. D. Houghton — 1
Department of Meteorology
University of Wisconsin
Madison, Wisconsin 53706

36. Dr. Joseph Huang — 1
Great Lake Environmental Res. Lab.
NOAA
2300 Washtenaw Avenue
Ann Arbor, Michigan 48104

37. Dr. W. R. Lambertson — 1
Fleet Numerical Weather Central
Naval Postgraduate School
Monterey, California 93940

38. Dr. C. E. Leith — 1
National Center for Atmospheric Research
P. O. Box 3000
Boulder, Colorado 80303

39. Dr. J. M. Lewis — 1
Laboratory for Atmospheric Research
University of Illinois
Urbana, Illinois 61801

40. Lieutenant Gary W. Ley — 1
2304 LaSalle Drive
Reading, Pennsylvania 19609

41. Dr. E. N. Lorenz — 1
Department of Meteorology
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

42. Dr. S. K. Kao — 1
Department of Meteorology
University of Utah
Salt Lake City, Utah 84112

43. Dr. A. Kasahara — 1
National Center for Atmospheric Research
P.O. Box 3000
Boulder, Colorado 80303

44. Dr. R. Madala — 1
Code 7750
Naval Research Laboratory
Washington, D. C. 20375

45. Dr. J. D. Mahlman — 1
Geophysical Fluid Dynamics Laboratory
Princeton University
Princeton, New Jersey 08540

46. Meteorology Library (Code 51) — 1
Naval Postgraduate School
Monterey, California 93940

47. Lieutenant William F. Mihok — 1
Fleet Numerical Weather Central
Naval Postgraduate School
Monterey, California 93940

48. Dr. S. Mudrick — 1
AFCRL (YD)
L. G. Hanscom Field
Bedford, Massachusetts 01730

49. National Center for Atmospheric Research — 1
Box 1470
Boulder, Colorado 80302

50. Office of Naval Research — 1
Department of the Navy
Washington, D. C. 20360

51. Director, Naval Research Laboratory — 1
ATTN: Technical Services Information Center
Washington, D. C. 20390

52. Dr. R. E. Newton 1
Department of Mechanical Engineering
Naval Postgraduate School
Monterey, California 93940

53. Dr. D. Nguyen 1
Department of Mechanical Engineering
Naval Postgraduate School
Monterey, California 93940

54. Dr. E. C. Nickerson 1
NOAA
Atmospheric Physics and Chemistry Laboratory
Boulder, Colorado 80302

55. Department of Oceanography, Code 58 1
Naval Postgraduate School
Monterey, California 93940

56. Dr. T. Ogura 1
Laboratory for Atmospheric Research
University of Illinois
Urbana, Illinois 61801

57. Professor K. Ooyama 1
National Center for Atmospheric Research
P.O. Box 3000
Boulder, Colorado 80303

58. Professor N. A. Phillips 1
National Meteorological Center/NOAA
World Weather Building
Washington, D. C. 20233

59. Dr. S. Piacsek 1
Code 7750
Naval Research Laboratory
Washington, D. C. 20390

60. Dr. R. J. Renard, Code 51Rd 1
Department of Meteorology
Naval Postgraduate School
Monterey, California 93940

61. Dr. T. Rosmond 1
Environmental Prediction Research Facility
Naval Postgraduate School
Monterey, California 93940

62. Dr. David Salinas 1
Department of Mechanical Engineering
Naval Postgraduate School
Monterey, California 93940

63. Dr. F. Sanders 1
Department of Meteorology
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

64. Dr. Y. Sasaki 1
Department of Meteorology
University of Oklahoma
Norman, Oklahoma 73069

65. Dr. A. Schoenstadt 1
Department of Mathematics
Naval Postgraduate School
Monterey, California 93940

66. Dr. Fred Shuman, Director 1
National Meteorological Center
World Weather Building
Washington, D. C. 20233

67. Dr. Joanne Simpson 1
Department of Environmental Sciences
2015 Ivy Road
Charlottesville, Virginia 22903

68. Dr. J. Smagorinsky, Director 1
Geophysical Fluid Dynamics Laboratory
Princeton University
Princeton, New Jersey 08540

69. Dr. R. Somerville 1
National Center for Atmospheric Research
P.O. Box 3000
Boulder, Colorado 80303

70. Dr. Peter H. Stone 1
Department of Meteorology
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

71. Dr. J. Wallace 1
Department of Atmospheric Sciences
University of Washington
Seattle, Washington 98105

72. Dr. D. Williamson 1
National Center for Atmospheric Research
P.O. Box 3000
Boulder, Colorado 80303

73. Dr. F. J. Winninghoff 1
1085 Steelse Avenue, #503
Willowdale (Toronto)
Ontario M2R2T1 Canada

74. Dr. M. G. Wurtele 1
Department of Meteorology
University of California
Los Angeles, California 90024

75. Dr. J. Young 1
Department of Meteorology
University of Wisconsin
Madison, Wisconsin 53706

Thesis
H5735
c.1

163161

Hinsman

Application of a
finite element method
to the barotropic
primitive equations.